

ADVERTIMENT. La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX (www.tesisenxarxa.net) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA. La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR (www.tesisenred.net) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING. On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX (www.tesisenxarxa.net) service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the TDX service. Introducing its content in a window or frame foreign to the TDX service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

PHD THESIS

Electromagnetic Models for Ultrasound Image Processing

Author:

Hugo NAVARRETE

Supervisor:

Ramón MUJAL, Ph.D

*A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy
in the*



Department of Electrical Engineering

November 2015

Declaration of Authorship

I, Hugo NAVARRETE, declare that this thesis titled, 'Electromagnetic Models for Ultrasound Image Processing' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“The same equations have the same solutions”

Richard Phillips Feynman
The Feynman Lectures on Physics Vol II

Acknowledgements

I am deeply grateful to my advisor Dr. Ramón Mual-Rosas, PhD for his helpful guidance and unconditional support. To AECI (Agencia Española de Cooperación Internacional) and COLCIENCIAS (Departamento Administrativo de Ciencia, Tecnología e Innovación, Colombia) for their financial support in the initial stages of the project. To my friend Dr Alejandro César Frery, PhD who first pointed out the basic ideas of this work. To my brother Dr. Solón Navarrete M.D. for his comments and fruitful discussions. And to My Wife, Mónica: you are part of my life, this could not have been done without your love and support.

Contents

Declaration of Authorship	iii
Acknowledgements	vii
Abstract	1
Prologue	3
Motivation	3
Objectives	3
Contributions	3
Thesis outline	4
1 Introduction	5
1.1 General Considerations	5
1.2 Central Limit Theorem Approach	6
1.3 Compound Representation Approach	7
1.4 Multiplicative Model Approach	8
1.5 Logarithmic compression	8
2 Statistical Models for Ultrasound RF data	11
2.1 Central Limit Theorem Approach	11
2.1.1 Rayleigh distribution	11
2.1.2 Rice distribution	12
2.1.3 K-Distribution	16
2.1.4 Homodyned K-Distribution	18
2.2 Compound representation approach	18
2.2.1 Homodyned K-Distribution	18
2.2.2 Generalized K-Distribution	30
2.2.3 Rician Inverse Gaussian Distribution (RiIG)	30
2.3 Multiplicative model approach	30
2.3.1 Generalized K-Distribution	31
3 A general model for ultrasound B-scan images	33
3.1 Statistical models for SAR data	33
3.1.1 Speckle Noise Model	33
3.1.2 The Multiplicative Model	34
3.2 G_A distributions for SAR images	35
3.2.1 K_A -distribution for SAR image	37
3.2.2 GA_0 distribution for SAR image	37
3.3 G_0 distribution for ultrasound image processing	38
3.4 HG_0 distribution for Log-compressed B-scan images	39
3.4.1 Logarithmic Compression Model	39
3.4.2 A New Statistical Model of Log-compressed B-scan images	40
3.5 Moments generating function method for HG_0 parameter estimation	41

3.6	Maximum likelihood method for HG0 parameter estimation	42
4	Log-compressed data modeling	43
4.1	Montecarlo simulation of Log-compressed data	43
4.2	Homodyned-K distributed data simulation	43
4.3	Stable process simulation	44
4.4	Log Compression	44
4.5	HG0 distribution for modeling Log-compressed data	44
4.6	Concluding Remarks	45
5	Central Limit Theorem Revisited	73
5.1	Generalized Central limit theorem	73
5.2	α -Stable distributions	73
5.2.1	Definition	74
5.3	Infinite divisibility and Alpha-stable distributions	74
5.4	Self-decomposable distributions	74
5.5	The Kullback–Leibler divergence	75
5.6	G0 and Alpha-stable distributions	75
5.7	The main theorem	75
5.8	Proof	75
5.8.1	Multiplicative Model and G0 distribution in Intensity format	75
5.8.2	G0 Multiplicative Model representation	76
5.8.3	G0 Compound representation	76
5.8.4	Shanbhag and Sreehari, 1977 theorem	76
5.8.5	Stable distributions in Amplitude format	78
5.8.6	Log-compression of stable distributions	78
5.9	The multiplicative model and the Generalized Central Limit Theorem	78
5.10	Compound representation and the multiplicative model	78
5.11	Concluding remarks	79
6	Applications	81
6.1	B-scan image filtering.	81
6.2	B-scan image segmentation	81
6.3	Heart ejection fraction estimation	81
7	Conclusions and future work	85
7.1	Conclusions	85
7.2	Future Work	85
A	Moments generating function for HG0 distribution	87
B	MATLAB code used in this work	89
B.1	Simulating and plotting K-distributed data	89
B.1.1	PDF and CDF evaluation	89
B.1.2	Plotting PDF and CDF	89
B.2	Simulating and plotting Homodyned K-distributed data	90
B.2.1	Simulating Homodyned-K random numbers	90
B.2.2	Montecarlo experiments	90
B.2.3	Polynomial fit for HG0 parameters estimation	91
B.2.4	Polynomial objective function for for HG0 parameters estimation	93
B.2.5	Parameters Estimation	93

B.2.6	Kullback-Leibler divergence	94
C	MATLAB Code for Stable Distributions	95
C.1	Stable Distribution PDF	95
C.2	Stable Distribution CDF	102
C.3	Stable Distributed Random Numbers	106
C.4	Stable Distribution data fitting	107
C.5	Stable Distribution CDF inversion	118
D	MATLAB Code for Applications	137
D.1	Speckle Filtering	137
D.2	Image Segmentation	138
	Bibliography	141

List of Figures

2.1	Rayleigh Distribution	13
2.2	Rice Distribution Motivation	14
2.3	Rice Distribution	15
2.4	K-distribution	17
2.5	Homodyned K Distributions ($\nu = 0, 0 < \alpha < 20$)	19
2.6	Homodyned K Distributions ($\nu = 0.5, 0 < \alpha < 20$)	20
2.7	Homodyned K Distributions ($\nu = 1.0, 0 < \alpha < 20$)	21
2.8	Homodyned K Distributions ($\nu = 2.0, 0 < \alpha < 20$)	22
2.9	Homodyned K Distributions ($\nu = 6.0, 0 < \alpha < 20$)	23
2.10	Homodyned K Distributions ($\alpha = 0.5, 0 < \nu < 20$)	24
2.11	Homodyned K Distributions ($\alpha = 1.0, 0 < \nu < 20$)	25
2.12	Homodyned K Distributions ($\alpha = 2.0, 0 < \nu < 20$)	26
2.13	Homodyned K Distributions ($\alpha = 4.0, 0 < \nu < 20$)	27
2.14	Homodyned K Distributions ($\alpha = 6.0, 0 < \nu < 20$)	28
2.15	Homodyned K Distributions ($\alpha = 16.0, 0 < \nu < 20$)	29
4.1	Montecarlo experiment 1	46
4.2	Montecarlo experiment 2	47
4.3	Montecarlo experiment 3	48
4.4	Montecarlo experiment 4	49
4.5	Montecarlo experiment 5	50
4.6	Montecarlo experiment 6	51
4.7	Montecarlo experiment 7	52
4.8	Montecarlo experiment 8	53
4.9	Montecarlo experiment 9	54
4.10	Montecarlo experiment 10	55
4.11	Montecarlo experiment 11	56
4.12	Montecarlo experiment 12	57
4.13	Montecarlo experiment 13	58
4.14	Montecarlo experiment 14	59
4.15	Montecarlo experiment 15	60
4.16	Montecarlo experiment 16	61
4.17	Montecarlo experiment 17	62
4.18	Montecarlo experiment 18	63
4.19	Montecarlo experiment 19	64
4.20	Montecarlo experiment 20	65
4.21	Montecarlo experiment 21	66
4.22	Montecarlo experiment 22	67
4.23	Montecarlo experiment 23	68
4.24	Montecarlo experiment 24	69
4.25	Montecarlo experiment 25	70
4.26	Montecarlo experiment 26	71
4.27	Montecarlo experiment 27	72

6.1	Ultrasound B-scan image filtering	82
6.2	Ultrasound B-scan image segmentation	83
6.3	Ejection Fraction estimation	84

Dedicated to My Parents María Elvia (in memoriam) and Jesús Francisco. To My Sisters and Brothers: José "Froidon" (in memoriam), Carlos Alberto "Lagüete", Carmenza "Mana", Marcelo "March", Aristóbulo "Lacrija", Marina "Palela", Clementina "Tina" (in memoriam), Solón "Ranitas", Jairo "Yaripal", Sandra "Lachanda" and Elkin "Burropín". To My Wife Mónica. And to my son, Hugo Alberto

Abstract

Speckle noise is inherent when coherent illumination is employed because the backscattered echoes from the randomly distributed scatterers in the microscopic structure, i.e. under spatial resolution, of the medium are the origin of speckle phenomenon; which characterizes coherent imaging with a granular appearance. It is the case for example of Laser, Synthetic Aperture Radar (SAR), Sonar, Magnetic Resonance, X-ray or Ultrasound imagery.

Due to its random nature, statistical modeling is of particular relevance when dealing with speckled data in order to obtain efficient image processing algorithms. It can be shown that speckle noise is of multiplicative nature, strongly correlated and more importantly, with non-Gaussian statistics. These characteristics differ greatly from the traditional assumption of white additive Gaussian noise, often assumed in image segmentation, filtering, and in general, image processing; which leads to reduction of the methods effectiveness for final image information extraction; therefore, this kind of noise severely impairs human and machine ability to image interpretation.

Clinical ultrasound imaging systems employ nonlinear signal processing to reduce the dynamic range of the input echo signal to match the smaller dynamic range of the display device and to emphasize objects with weak backscatter. This reduction in dynamic range is normally achieved through a logarithmic amplifier i.e. logarithmic compression, which selectively compresses large input signals. This kind of nonlinear compression totally changes the statistics of the input envelope signal; and, a closed form expression for the density function of the logarithmic transformed data is usually hard to derive.

This thesis is concerned with the statistical distributions of the Log-compressed amplitude signal in coherent imagery, and its main objective was to develop a general statistical model for log-compressed ultrasound B-scan images. The developed model is adapted, making the pertinent physical analogies, from the Multiplicative Model in Synthetic Aperture Radar (SAR) context. It is shown that the proposed model can successfully describe log-compressed data generated via Montecarlo methods from models proposed in the specialized ultrasound image processing literature. Also, the model is successfully applied to model in-vivo echo-cardiographic (ultrasound) B-scan images.

Necessary theorems are established to account for a rigorous mathematical proof of the validity and generality of the model. Additionally, a physical interpretation of the parameters is given, and the connections between the generalized central limit theorem, the multiplicative model and the compound representations approaches for the different models proposed up-to-date, are established. It is also shown that the log-amplifier parameters are included as model parameters and all the model parameters are estimated using moments and maximum likelihood methods. Finally, three applications are developed: speckle noise identification and filtering; segmentation of in vivo echo-cardiographic (ultrasound) B-scan images and a novel approach for heart ejection fraction evaluation.

Keywords. Statistical Image Processing, Ultrasound B-scan Images, Speckle noise filtering, Generalized Central Limit Theorem, Alpha Stable Distributions, Log-Compressed Data Distribution, Random walks, Heart Ejection Fraction measurement, Echo-cardiographic image segmentation.

Prologue

Motivation

Various models have been introduced in the literature during last forty years for the statistics of ultrasound echo signals. However, when a log-compression or other (non-linear or linear) operators are applied to the echo envelope, the distribution of gray levels no longer follows the distributions computed for the RF echo envelope; as a result, computing the Log-compressed data distribution remains as an open problem whose solution should be applied to a wide kind of both theoretical and practical problems, with a high potential for patents generation processes.

Objectives

- The main objective is to develop a general model that includes the known models as particular cases.
- Once the general model is obtained, the objective is apply it to ultrasound B-scan image processing.

Contributions

- A new statistical model for Ultrasound B-scan images is developed.
- It is shown that the new model includes as a particular cases, after Log-compression, most of the models proposed up-to-date .
- The new model parameters include the Log-amplifier parameters, and can be estimated with standard parameter estimation methods.
- The new model is applied successfully to:
 - Theoretical random walk and Brownian stochastic process study.
 - Speckle noise identification and filtering.
 - Ultrasound B-scan image segmentation.
 - Heart ejection fraction (EF) evaluation

Thesis outline

- Chapter 1: Introduction
A review of previous and related works, state of the art.
- Chapter 2: Statistical models for Ultrasound RF data
A review of the models proposed up-to-date for RF envelope data is presented. Also, different approaches for study those distributions are presented and related: Central Limit Theorem, Compound Distribution and Multiplicative model approaches.
- Chapter 3: A general model for ultrasound B-scan images
GA0 from SAR image processing is studied and adapted to ultrasound RF envelope data. Then, its Log-Compressed version and the physical meaning of model parameters are deduced. Also, model parameters are estimated using moments and maximum likelihood methods.
- Chapter 4: Log-compressed data modeling.
A Montecarlo study is performed with the distributions presented in Chapter 3. Log-compressed data are simulated and then they are modeled with the Model developed in Chapter 4. Finally, goodness of fit tests are performed for null hypothesis testing.
- Chapter 5: Central Limit Theorem revisited.
The mathematical theory is presented to demonstrate the model generality. It is shown formally that the developed model has as particular cases all the models presented in chapter 3, after Log-compression.
- Chapter 6: Applications
 - Speckle noise filtering in Log-compressed B-scan images.
The new model is applied to speckle noise identification and adaptive filtering.
 - Log-compressed B-scan images segmentation.
The new model is applied to Log-compressed B-scan images segmentation.
 - Heart ejection fraction (EF) measurement.
The importance of EF estimation is presented and the new model is applied for EF measurement. A comparison study is carried to verify the performance of the new approach.
- Chapter 7: Conclusions and future work.
General guidelines for applications and extensions of the new model are exposed.

Chapter 1

Introduction

1.1 General Considerations

Speckle noise is inherent to coherent imaging systems because the backscattered echoes from the randomly distributed scatterers in the microscopic structure, i.e. under spatial resolution, of the medium are the origin of speckle phenomenon; which characterizes coherent imaging with a granular appearance. It is the case for example of Laser, Synthetic Aperture Radar (SAR), Sonar, Magnetic Resonance, X-ray or Ultrasound imagery.

The emitted or interrogating pulse is modeled as a superposition of progressive plane waves; when these waves encounter an interface between two media with different wave propagation properties, a part of the incident waves are reflected (specular echoes). Along with these specular echoes, backscattered echoes from the microscopic structure of the medium are added.

Due to its random nature, statistical modeling is of particular relevance when dealing with speckled data in order to obtain efficient image processing algorithms. Speckle noise is of multiplicative nature, strongly correlated and more importantly, with non-Gaussian statistics. These characteristics differ greatly from the traditional assumption of white additive Gaussian noise, often assumed in image segmentation, filtering, and in general, image processing; which leads to reduction of the methods effectiveness for final image information extraction; therefore, this kind of noise severely impairs human and machine ability to image interpretation. Several distribution families have been proposed for coherent illuminated images. The large variability of models is due to the strong dependence of the observed statistics on the density of scatters and on their spatial distribution. Most of the reported models are for describe the envelope of received echo signal; but in the ultrasound case it is necessary to account for particular signal processing transformations. This means, that the proposed distributions are valid before typical transformations like low-pass filtering, interpolation, log-compression and Time-Gain-Compensation; in consequence, they are not longer valid for the ultrasound images acquired under clinical conditions

In this thesis, the various models for the first-order statistics of the echo envelope found in the literature are first presented, based on their representation using the multiplicative model, the central limit theorem and the compound representation approaches. Then, these models are Log-compressed. This transformation led usually to non-analytical expressions which makes necessary the use of Montecarlo simulation methods in order to study the feasibility of any statistical description of the Log-compressed data.

Another aspect of a statistical distribution is the physical meaning of the model parameters in order to relate its goodness of fit on real data with the properties of the analyzed media. It is the case in problems such as tissue characterization (Shankar et al., 1993; Shankar, 2001; Shankar et al., 2001; Oelze, O'Brien, and Zachary, 2007; Tsui et al., 2008; Vegas-Sanchez-Ferrero et al., 2012) where estimated parameters themselves

are used for classifying purposes. In such a framework, it is necessary to pay attention to the physical meaning of the distribution parameters. For instance, a mixture of sufficiently large number of Gaussian distributions could model the histogram of echo envelope, but the physical meaning of the statistics proportions of such mixtures is not clear because a Gaussian distribution is not (directly) meaningful for the first-order statistics of the echo envelope.

The backscattered echo signal received at the transducer of an ultrasound device can be viewed as the vector sum of the individual signals produced by the scatterers distributed in the medium (Wagner et al., 1983; Wagner, Insana, and Brown, 1987). As a result, the framework leading to a physical interpretation of the statistical distributions parameters assumes that the individual contributions of the scatterers are independent. When exists a periodicity pattern in the scatterers positions (Wagner et al., 1983; Wagner, Insana, and Brown, 1987), or strong specular reflections, then a coherent or deterministic component appears in the received signal, because of a scatterers long-range organization (relative to the wavelength). The power of the coherent component is called the coherent signal power. The remaining power (from the total signal power) is called the diffuse signal power and corresponds to the diffuse (or random) component, made of a diffuse collection of scatterers.

1.2 Central Limit Theorem Approach

Central Limit Theorem states that the sum of a independent and identically distributed (i.i.d.) random variables with finite variances will tend to normal distribution as the number of variables grows. As a result, the magnitude of a limit random phasor sum with finite mean and variance is Rayleigh distributed. Rayleigh distribution was first introduced in the context of sound propagation in Rayleigh, 1880. In ultrasound imaging, the Rayleigh distribution corresponds to the distribution of the gray level (also called amplitude) in an unfiltered B-mode image, viewed as the envelope of the radio-frequency (RF) image, in the case of a high effective density of random scatterers with no coherent signal component (Wagner et al., 1983).

The Rice distribution also corresponds to a high effective density of random scatterers (the diffuse signal component), but combined with the presence of a coherent signal component of power. Thus, the Rayleigh distribution is the special case of the Rice distribution, with no coherent component. The Rice distribution itself first appeared in the context of wave propagation (Nakagami, 1960; 1940; Rice, 1945); in ultrasound was applied by Insana, 1986; Wagner; 1987 and Tuthill, Sperry, and Parker, 1988.

The K-distribution corresponds to a variable (effective) density α of random scatterers, with no coherent signal component and was introduced in ultrasound imaging by Shankar et al., 1993. The parameter α can be viewed as the number of scatterers per resolution cell or “density”, multiplied by a coefficient depending on the scanning geometry and parameters, and the backscatter coefficient statistics, i.e. “effective” (Jakeman and Pusey, 1976 and Shankar et al., 1993). The parameter α is also called the scatterer clustering parameter Dutt and Greenleaf, 1994. The distribution itself appeared first in Lord, 1954 in the context of random walks, and was further studied by Jakeman and Pusey, 1976 in the context of sea echo.

The homodyned K-distribution corresponds to the general case of a variable effective density of random scatterers with or without a coherent signal component introduced in ultrasound by Dutt and Greenleaf, 1994. The Homodyned K-distribution was first introduced and studied in Jakeman, 1980 and Jakeman and Tough, 1987 in the context

of random walks viewed as a model of weak scattering. Thus, K-distribution, Rice and Rayleigh are particular and/or limiting cases (namely, the effective density α of random scatterers is “infinite”) of the Homodyned K-distribution, which may be considered the general model.

A Central Limit Theorem Generalization due to Gnedenko and Kolmogorov (Hoeffding et al., 1955) states that the sum of a number of random variables with a power-law tail (Paretian tail with power α) distribution (and therefore having infinite variance) will tend to a Alpha-Stable distribution as the number of summands grows. If the exponent $\alpha > 2$ then the sum converges to a stable distribution with stability parameter equal to 2, i.e. a Gaussian distribution. As a result Rayleigh distribution is a special case of the square-root-symmetric stable distribution. In Pereyra and Batatia, 2012, Alpha-stable distributions have been applied to statistical image processing of high frequency ultrasound imaging, in order to perform tissue segmentation in ultrasound images of skin. It was established that ultrasound signals backscattered from skin tissues converge to a complex Levy Flight random process with non-Gaussian-stable statistics. Based on these results, it was proposed to model the distribution of multiple-tissue ultrasound images as a spatially coherent finite mixture of heavy-tailed Rayleigh distributions i.e. alpha-stable distributions.

1.3 Compound Representation Approach

One important result of Jakeman and Tough, 1987 is that the Homodyned K-distribution admits a compound representation, i.e. the distribution can be viewed as the marginal distribution of a model in which the Rice distribution has its diffuse signal power modulated by a gamma distribution with mean μ and variance α . In other words, given the effective density of random scatterers, the model results from the joint probability of the amplitude and the modulating variable w (distributed according to a gamma distribution), and the marginal distribution of the variable A is obtained by integrating the joint probability over the domain of w . In the same manner, the K-distribution is the marginal distribution of a model in which the Rayleigh distribution has its diffuse signal power modulated by a gamma distribution.

Another modeling possibility, introduced in Barakat, 1986 and further developed in Jakeman and Tough, 1987, is equivalent to modulate both the coherent signal component and the diffuse signal power of the Rice distribution by a gamma distribution giving rise to the generalized K-distribution. This distribution has been used in ultrasound imaging in Eltoft, 2006c. However, in Eltoft, 2005, the Rician inverse Gaussian distribution (RiIG) is introduced, and it corresponds to a model in which both the coherent signal component and the diffuse signal power of a Rice distribution are modulated by an inverse Gaussian (IG) distribution, instead of a gamma distribution.

The homodyned K-distribution, the generalized K-distribution and the RiIG are distributions with three parameters: two parameters for the modulated Rice distribution and one parameter for the modulating (gamma or IG) distribution. A simpler model consists in modeling the gray level of the speckle pattern in a B-mode image by a Nakagami distribution (Shankar, 2000). The Nakagami distribution is a two-parameter distribution first introduced in Nakagami, 1960 in the context of wave propagation. It can be viewed as an approximation of the Homodyned K-distribution, at least in the special cases of the Rice distribution and the K-distribution. That what essentially the point of view of Nakagami, 1960 in the context of random walks and wave propagation, although the homodyned K-distribution was not yet introduced.

Three other distributions were introduced in the context of ultrasound imaging. The first one is called the generalized Nakagami distribution (Shankar, 2001) and is obtained from the Nakagami distribution by a change of variable. This distribution was also proposed independently in Raju and Srinivasan, 2002 (in the equivalent form of a generalized gamma distribution). The second other distribution is called the Nakagami-gamma (NG) distribution (Shankar, 2003). That distribution can be viewed as the marginal distribution of a model in which the Rice distribution is approximated by a Nakagami distribution, and in which its total signal power (that would correspond to the total signal power of the Rice distribution) is modulated by a gamma distribution. Equivalently, the corresponding Rice distribution would have both its coherent signal power and its diffuse signal power modulated by the gamma distribution. The third distribution is called the Nakagami-generalized inverse Gaussian (NGIG) distribution (Agrawal and Karmeshu, 2007), and it corresponds to a model in which the (approximating) Nakagami distribution has its total signal power modulated by a generalized inverse Gaussian (GIG) distribution instead of a gamma distribution.

1.4 Multiplicative Model Approach

The statistical modeling of Synthetic Aperture Radar (SAR) imagery has provided some of the best tools for the processing and understanding of coherent imaging. Among the statistical approaches the most successful is the multiplicative model. This model offers a set of distributions that, with a few parameters, are able to characterize most of SAR data. This model is presented, for instance, in Oliver and Quegan, 2004, and extended in Frery et al., 1997. This extension is a general and tractable set of distributions within the multiplicative model, used to describe every kind of SAR return. It was then called a universal model, and its properties are studied in Frery et al., 1999; Mejail et al., 2000; Mejail et al., 2003. The multiplicative model is based on the assumption that the observed random field Z is the result of the product of two independent and unobserved random fields: X and Y . The random field X models the terrain backscatter, and thus depends only on the type of area each pixel. The random field Y describes the speckle noise, taking into account that the effective number of looks L (ideally) independent images are averaged in order to reduce the noise. There are various ways of modelling the random fields X , whereas the physics of speckle noise allows the assumption of a law for Y . This universal model proposes the distribution to describe the amplitude backscatter X , yielding the GA distribution for the return, and the K-distribution is a particular case of this model; while in Eltoft, 2006a the Generalized-K and Homodyned-K distributions can be represented as a convolution of these models, i.e. a scale mixture of Gaussians models.

1.5 Logarithmic compression

It should be noted that the distributions mentioned before concern the envelope of the RF signal, but when a log-compression or other (nonlinear or linear) operators are applied to the envelope, the distribution of the gray levels no longer follows the distributions computed for the RF echo envelope. In the case of log-compression, the resulting distribution has been modeled in Dutt, 1995 and Dutt, 1996, assuming the K-distribution for rf echo envelope. In Prager et al., 2003, a decompression algorithm is proposed, assuming the homodyned K-distribution for the envelope. As mentioned

before, operators other than log-compression can be applied on the envelope. In Navarrete et al., 2002 the Multiplicative model was applied to in vivo B-scan images filtering; and in Nillesen et al., 2008, a linear filter was applied to the RF data before computing the envelope. Five distributions were tested to fit the data: the Rayleigh distribution, the K-distribution, the Nakagami distribution, the inverse Gaussian distribution and the gamma distribution. The authors showed, based on empirical tests that the gamma distribution best fit the data, although the physical meaning for the parameters was not established.

This thesis is concerned with the statistical distributions of the Log-compressed RF amplitude image, and therefore all distributions will be Log-compressed previously to obtain the filtered B-scan mode image version of the data, with special attention to the physical meaning of the parameters. Note that it should not be confused the distribution of the (filtered/compressed) B-scan mode amplitude image, with the distribution of the intensity (i.e., the square of the amplitude) of the (unfiltered) rf envelope image, which may be approximated with the Nakagami distribution of the (unfiltered) rf-envelope amplitude image (Shankar, 2000).

Chapter 2

Statistical Models for Ultrasound RF data

As stated previously, speckle noise is a inherent phenomenon present when coherent illumination is employed, as is the case of sonar, laser, ultrasound, Magnetic Resonance, X ray or and synthetic aperture radar (SAR) imagery; therefore, any statistical description of data from these areas can be adapted to the others taking care of the parameters physical meaning in each case. Also, it is necessary to take into account that when the product of the wave number with the mean size of the scatterers is much smaller than the wavelength, and the acoustic impedance of the scatterers is close to the impedance of the embedding medium, a high density of scatterers results in a packing organization that implies a correlation between the individual signals produced by the scatterers (Hawley, Kays, and Twersky, 1967; Twersky, 1975; 1978, 1987, 1988; Lucas and Twersky, 1987; Berger, 1991) Apart from that case, the backscattered echo signal received at the transducer of an ultrasound device is viewed as a random phasor sum of the individual signals produced by the scatterers distributed in the medium (Wagner et al., 1983; 1987). Therefore, statistical description of the envelope signal can be viewed as a statistical description of a random walk in the complex plane.

In this chapter, based on the two excellent reviews by Destremes and Cloutier, 2010 and Mamou and Oelze, 2013 chapter 10, a review of the models proposed up-to-date for RF envelope data is presented. Also, different approaches for study those distributions are presented and related: Central Limit Theorems, Compound Distribution and Multiplicative model approaches.

2.1 Central Limit Theorem Approach

2.1.1 Rayleigh distribution

Consider a random walk in the complex plane:

$$\mathbf{A} = \sum_{j=1}^N \mathbf{a}_j \quad (2.1)$$

\mathbf{A} is the resultant phasor after N steps. The Central Limit Theorem states that when N tends to infinity and the following conditions are met:

- Random phasors \mathbf{a}_j are independent i.e. uncorrelated.
- Phase and amplitude are independent.
- Phase is uniform distributed.

- Phasor amplitudes are independent and identically distributed, with mean μ and variance σ finite.

Then in the random walk of equation 2.1, the join distribution of the uncorrelated real A_r and imaginary A_i parts of the random phasor \mathbf{A} will be Gaussian with no correlation terms:

$$P(A_r, A_i) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{A_r^2 + A_i^2}{2\sigma^2}\right) \quad (2.2)$$

The envelope signal is the magnitude of resultant complex phasor:

$$X = \sqrt{A_r^2 + A_i^2} \quad (2.3)$$

Whose join density in polar coordinates can be expressed as:

$$P_x(X, \theta) = \frac{X}{2\pi\sigma^2} \exp\left(-\frac{X^2}{2\sigma^2}\right) \quad (2.4)$$

As a result, the Amplitude probability density function is the marginal density:

$$\begin{aligned} P_x(X) &= \int_{-\pi}^{\pi} P_x(X, \theta) d\theta = \int_{-\pi}^{\pi} \frac{X}{2\pi\sigma^2} \exp\left(-\frac{X^2}{2\sigma^2}\right) d\theta \\ P_x(X) &= \frac{X}{\sigma^2} \exp\left(-\frac{X^2}{2\sigma^2}\right) \end{aligned} \quad (2.5)$$

This distribution function is known as a Rayleigh distribution function, and it is depicted in figure 2.1. The parameter

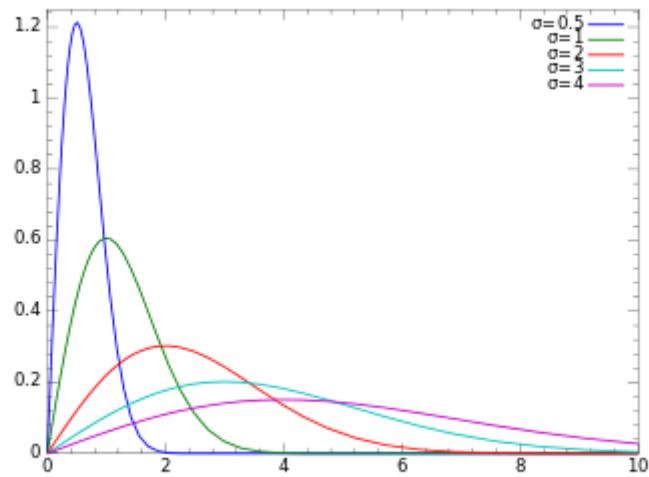
$$\sigma^2 = \frac{\bar{a}^2}{2}$$

where \bar{a}^2 is the mean intensity of the random step a_j before scaling by the factor $\frac{1}{\sqrt{N}}$. After normalizing the contribution of N independent scatterers by the factor $\frac{1}{\sqrt{N}}$, one also obtains \bar{a}^2 as the mean intensity (in fact, before or after taking the limit as $N \rightarrow \infty$). The idea behind the normalization factor of $\frac{1}{\sqrt{N}}$ (instead of $1/N$) is to average out the intensity of the scatterers (rather than their amplitude), to preserve the mean intensity. In the case of the Rayleigh distribution, the mean intensity (i.e., $2\sigma^2$) can be interpreted as the diffuse signal power, because there is no coherent component in the signal.

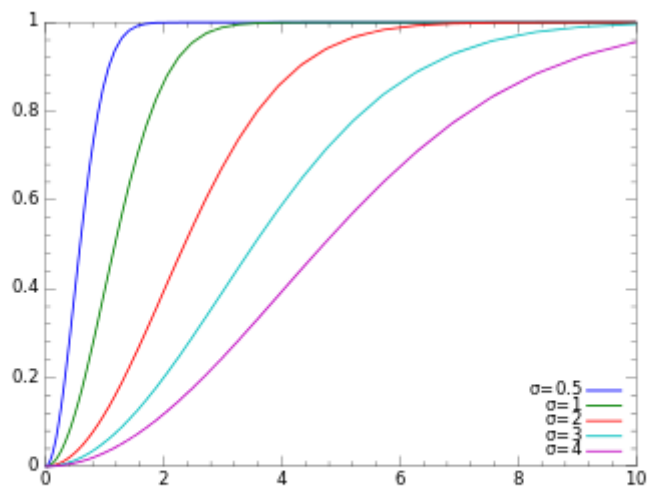
2.1.2 Rice distribution

Consider a random walk, obtained by adding a constant phasor ν to the random walk of equation 2.1 (see fig. 2.2) which may arise due to periodically located scatterers or due to strong specular scattering:

$$\mathbf{A} = \nu + \sum_{j=1}^N a_j \quad (2.6)$$



(A)



(B)

FIGURE 2.1: Rayleigh Probability Density Function (A) and Cumulative Distribution Function (B)

by Krishnavedala Own work. Licensed under CC0 via Commons

https://commons.wikimedia.org/wiki/File:Rayleigh_distributionPDF.svg#/media/File:Rayleigh_distributionPDF.svg

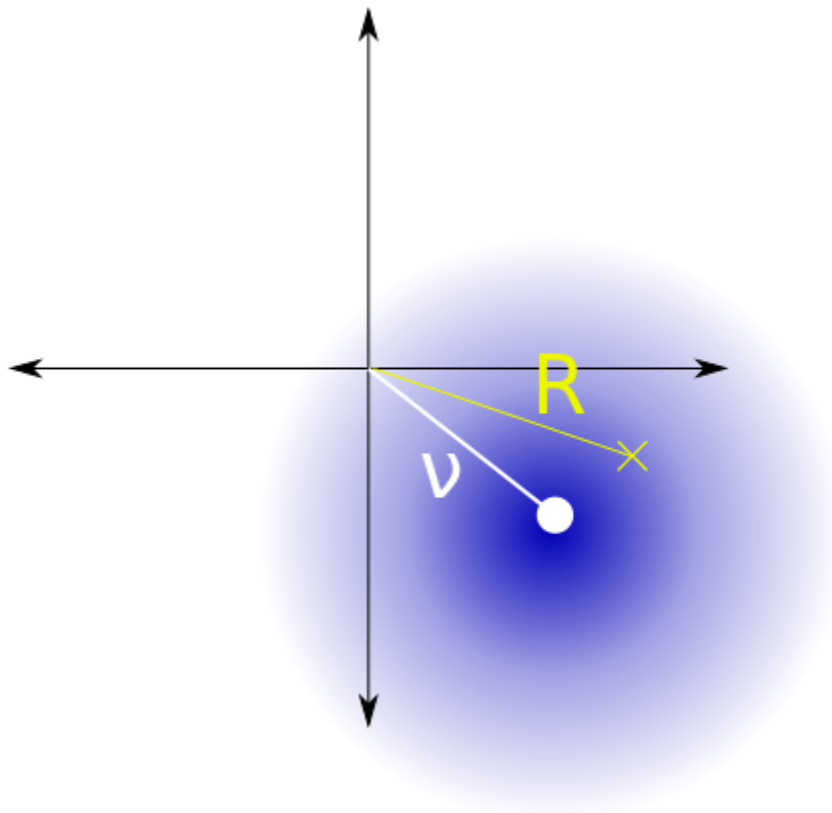


FIGURE 2.2: In the 2D plane, Rice distribution arises by picking a fixed point at distance ν from the origin. Generate a distribution of 2D points centered around that point, where the x and y coordinates are chosen independently from a gaussian distribution with standard deviation σ (blue region). If R is the distance from these points to the origin, then R has a Rice distribution.

by Sbyrnes321 - Own work. Licensed under CC0 via Commons -

https://commons.wikimedia.org/wiki/File:Rice_distribution_motivation.svg#/media/File:Rice_distribution_motivation.svg

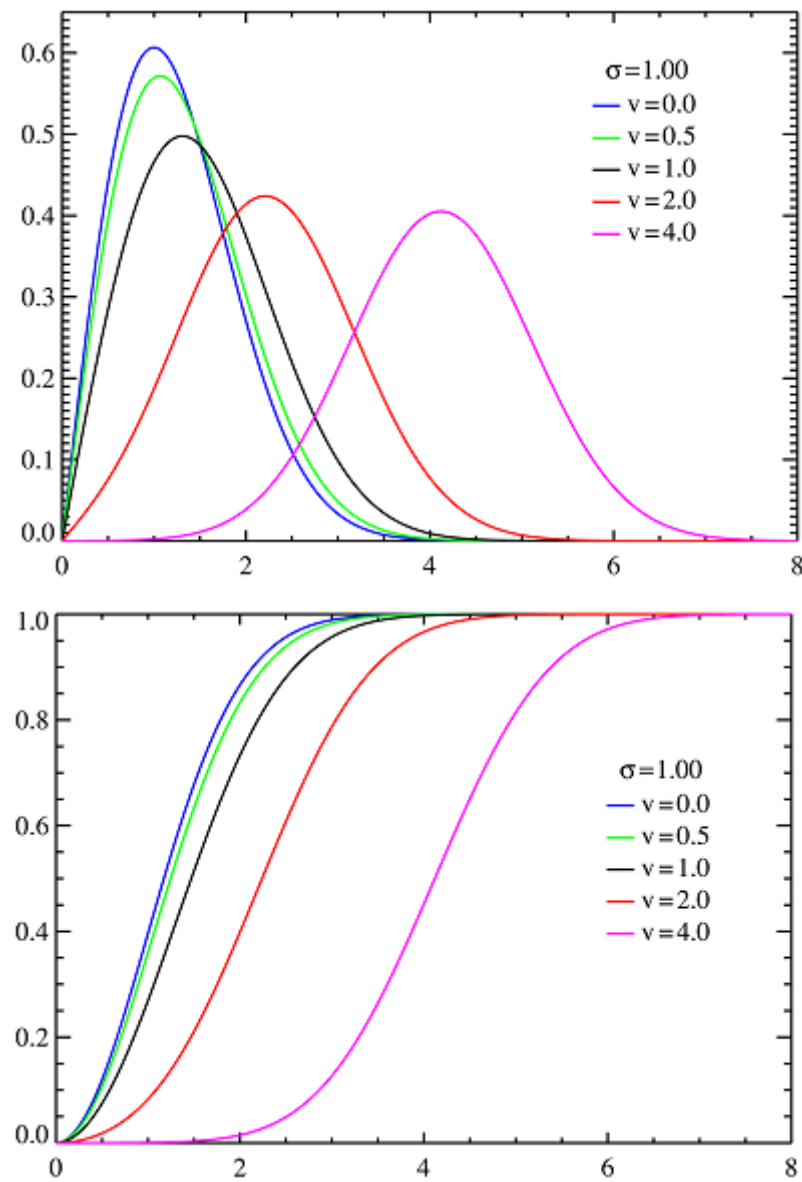


FIGURE 2.3: Rice Probability Density distribution (up) and its Cumulative Distribution Function (bottom)

Licensed under CC BY-SA 3.0 via Commons -

https://commons.wikimedia.org/wiki/File:Rice_distributiona_CDF.png#/media/File:Rice_distributiona_CDF.png

The joint density function of real and imaginary parts of \mathbf{A} can now be written as:

$$P(A_r, A_i) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(A_r + \nu_r)^2 + (A_i + \nu_r)^2}{2\sigma^2}\right) \quad (2.7)$$

The marginal density, can be expressed as (see Jakeman and Tough, 1987):

$$P_x(X) = \frac{X}{\sigma^2} \exp\left(-\frac{X^2 + \nu^2}{2\sigma^2}\right) I_0\left(\frac{\nu X}{\sigma^2}\right) \quad (2.8)$$

$I_0(x)$ denotes the modified Bessel function of the first kind of order 0 (the intensity I should not be confused with the Bessel function I_0). Figure 2.3 shows the Rice probability density function and its correspondent cumulative distribution function for different conditions of coherent component (ν) and fixed (σ). The figure illustrates that the Rice distribution approaches to Rayleigh distribution when the ratio of the coherent to diffuse ($r = \frac{\nu}{\sigma}$) power signals tends to 0; and when r tends to infinite, Rice distributions becomes a Gaussian Distribution.

2.1.3 K-Distribution

Consider the random walk described by equation 2.1, with independent phase and amplitude, and a uniformly distributed phase, in which the number of steps is variable. Namely, given α effective scatters per resolution cell and assuming that the number of steps N follows a negative binomial distribution of mean \bar{N} so that $p = \frac{1}{(1+\bar{N}/\alpha)}$. Let the random step be scaled by the factor $1/\sqrt{\bar{N}}$; then, the following random process is obtained:

$$N \sim \text{NegBin}(\alpha, 1/(1 + \bar{N}/\alpha))$$

$$A|N \sim \frac{1}{\sqrt{\bar{N}}} \sum_{j=1}^N a_j \quad (2.9)$$

Let \bar{N} tend to infinite in the random process of equation 2.9; then, the amplitude probability density function is a K-distribution (Jakeman and Tough, 1987):

$$P(X) = \frac{4X^\alpha}{(2\sigma^2)^{(\alpha+1)/2}} K_{\alpha-1}\left(\sqrt{\frac{2}{\sigma^2}}X\right), \quad \alpha, \sigma, X > 0 \quad (2.10)$$

K_ρ is the modified Bessel function of second kind and order ρ . The scale parameter

$$\sigma^2 = \bar{a}^2/(2\alpha) \quad (2.11)$$

is now divided by the effective scatters density per resolution cell, and the limit distribution when $\alpha \rightarrow \infty$ is a Rayleigh distribution; i.e. when the scatters per resolution cell is high, a completely developed speckle is obtained. It is also noted that the mean intensity $\bar{a}^2 = 2\alpha\sigma^2$ corresponds to the diffuse power signal of one scatterer. K-distribution is depicted in figure 2.4 showing the limit distribution as $\alpha \rightarrow \infty$, namely the Rayleigh distribution.

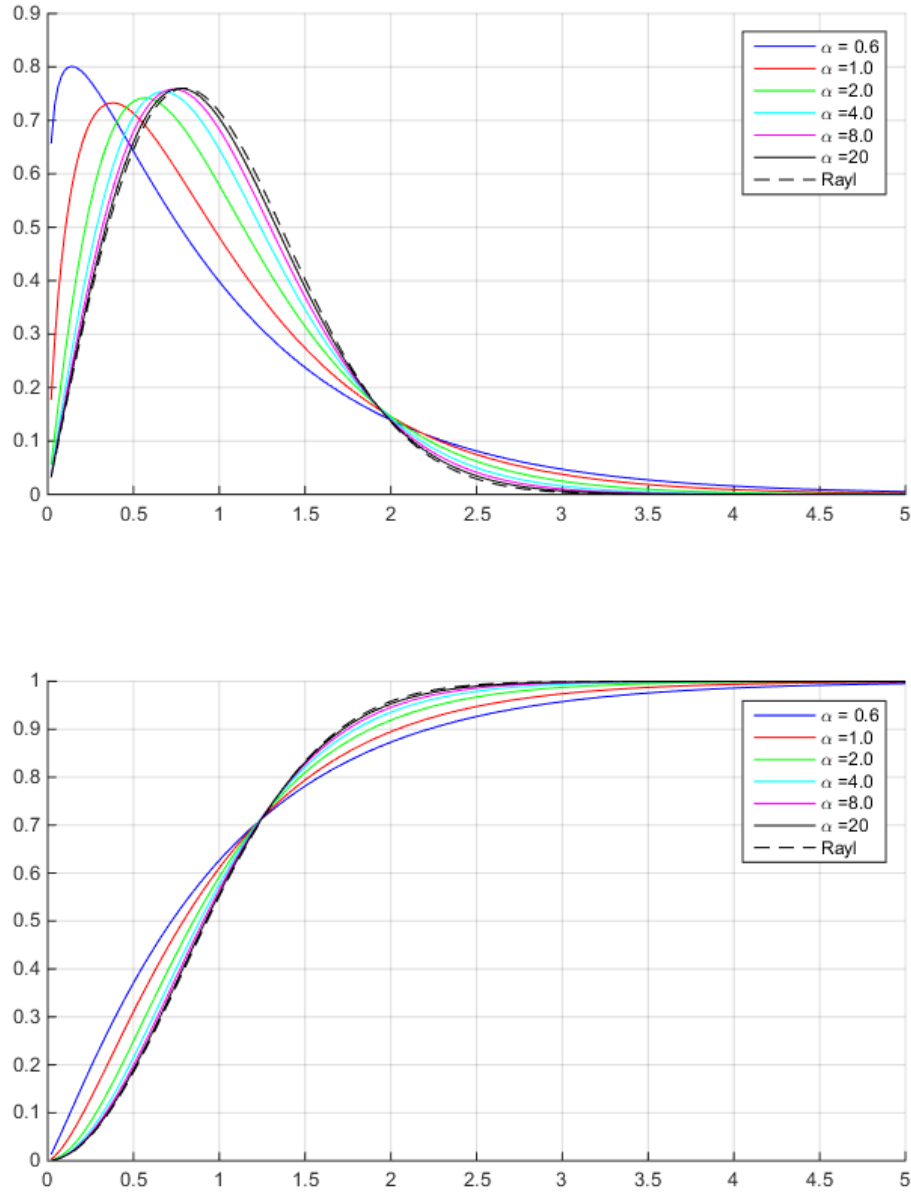


FIGURE 2.4: Probability Density K-Distributions (up) and its Cumulative Distribution Function (bottom). For different α values, scale parameter is adjusted for unitary mean. Rayleigh distribution with unitary mean is plotted as reference. Note that for $\alpha > 20$ K-distribution can be approximated with a Rayleigh distribution

2.1.4 Homodyned K-Distribution

Homodyned K-Distribution results from adding a coherent signal ν to the random process described by equation 2.9,

$$N \sim \text{NegBin}(\alpha, 1/(1 + \bar{N}/\alpha))$$

$$A|N \sim \nu + \frac{1}{\sqrt{N}} \sum_{j=1}^N a_j \quad (2.12)$$

Let \bar{N} tend to infinite in the random process of equation 2.12; then, the amplitude probability density function is a Homodyned K-distribution (Jakeman and Tough, 1987):

$$P(X|\sigma^2, \nu, \alpha) = X \int \frac{u \text{Jo}(u\nu) \text{Jo}(uX)}{\left(1 + \frac{u^2 \sigma^2}{2}\right)^\alpha}, \quad \alpha, \sigma, X > 0; \quad \nu \geq 0 \quad (2.13)$$

Jo denotes the Bessel Function of the First Kind and order 0; and scale parameter

$$\sigma^2 = \bar{a}^2/(2\alpha) \quad (2.14)$$

Homodyned-K distribution has no analytical expression, but can be represented in power series (see Dutt, 1995). Also, note that the Homodyned-K distribution when $\nu=0$ is the K-distribution and it is shown in Jakeman and Tough, 1987 that the limit distribution as $\alpha \rightarrow \infty$ is the Rice distribution.

2.2 Compound representation approach

Using characteristic function Jakeman and Tough, 1987 showed that random walks in n dimensions can be described using a compound representation or, equivalently, by the multiplicative model. In this section different proposals using the compound representation are presented.

2.2.1 Homodyned K-Distribution

Equation 2.15 bellow is the Homodyned-K distribution compound representation (see Jakeman and Tough, 1987):

$$P(X|\sigma^2, \nu, \alpha) = \int_0^\infty P_r(X|\nu, \sigma^2 w) P_\gamma(w|\alpha, 1) dw, \quad \alpha, \sigma, X > 0; \quad \nu \geq 0 \quad (2.15)$$

P_r is the Rice distribution with coherent component ν and variance $w\sigma^2$; and P_γ is the Gamma Distribution with mean and variance equal to α ; therefore, Homodyned-K distribution can be understood as a Rice distribution whose variance is a gamma-distributed random variable. This compound representation led to following limiting distributions:

- K- Distribution given $\nu \rightarrow 0$.
- Rice Distribution as $\alpha \rightarrow \infty$
- Rayleigh Distribution when $\alpha \rightarrow \infty, \nu \rightarrow 0$

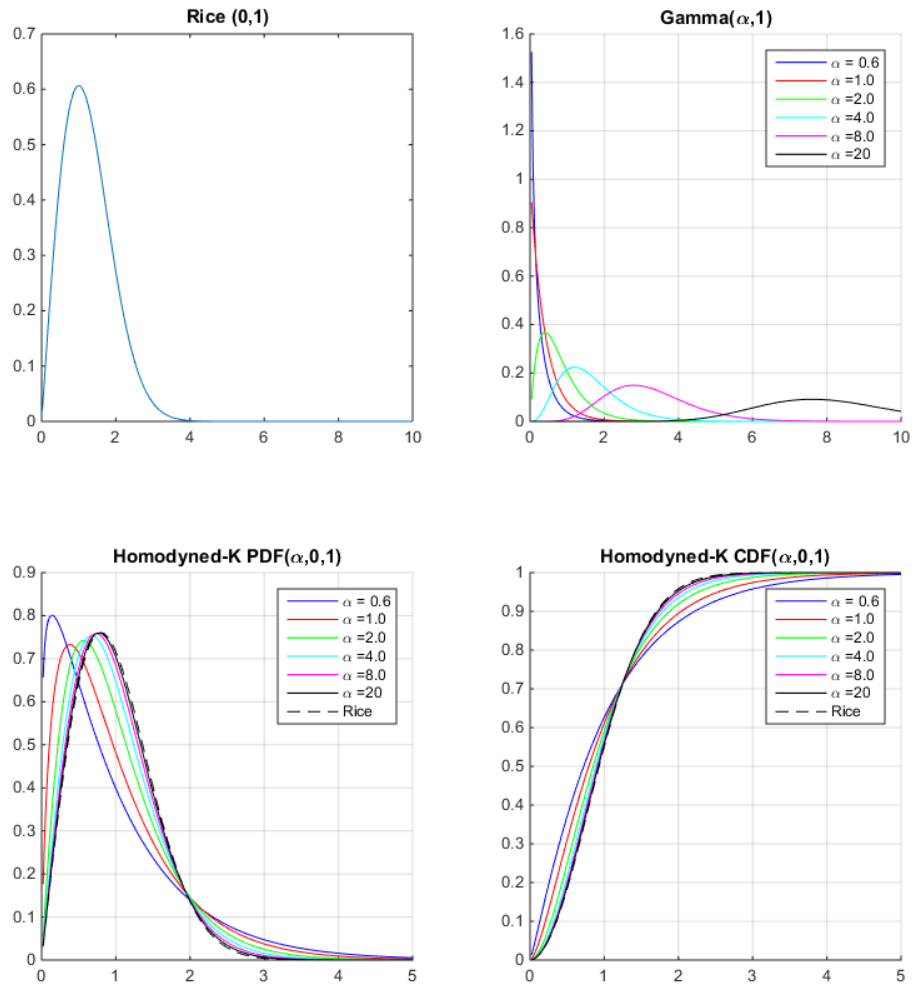


FIGURE 2.5: Homodyned K-distributions resulting of compounding a modulated Rayleigh ($\nu = 0$) with different effective scatter densities ($0.5 \leq \alpha \leq 20$) modulating Gamma distribution.

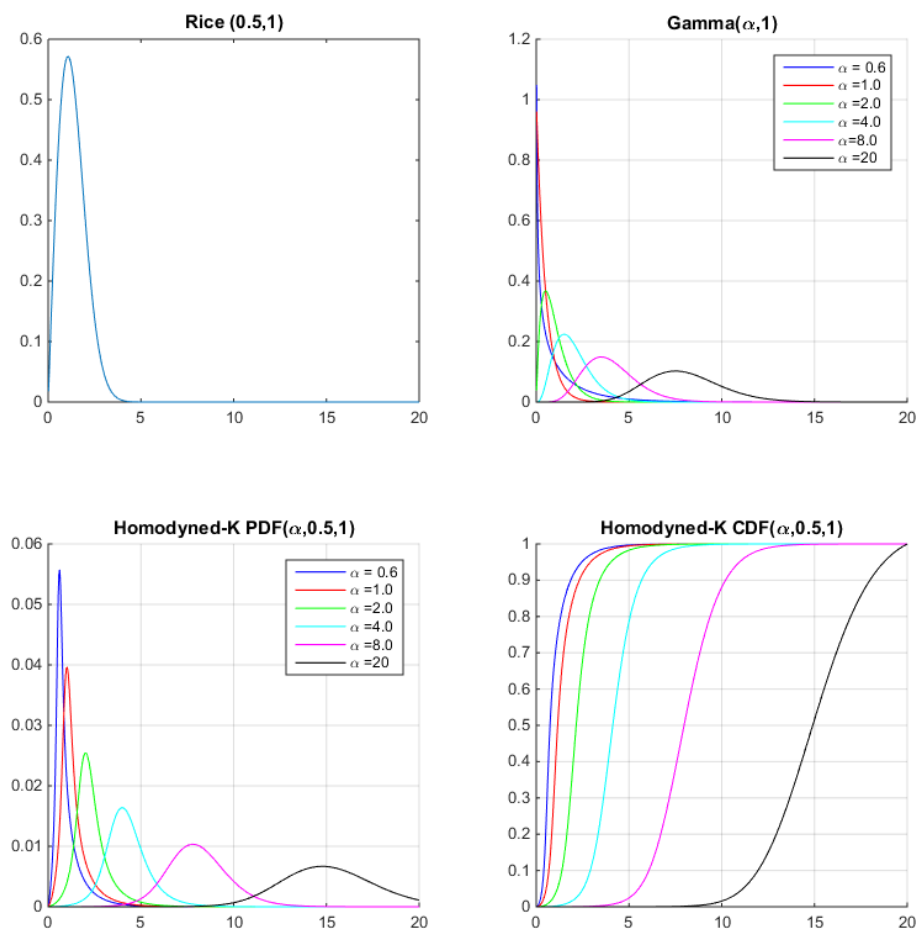


FIGURE 2.6: Homodyned K-distributions resulting of compounding a modulated Rice ($\nu = 0.5$) with different effective scatter densities ($0.5 \leq \alpha \leq 20$) modulating Gamma distribution.

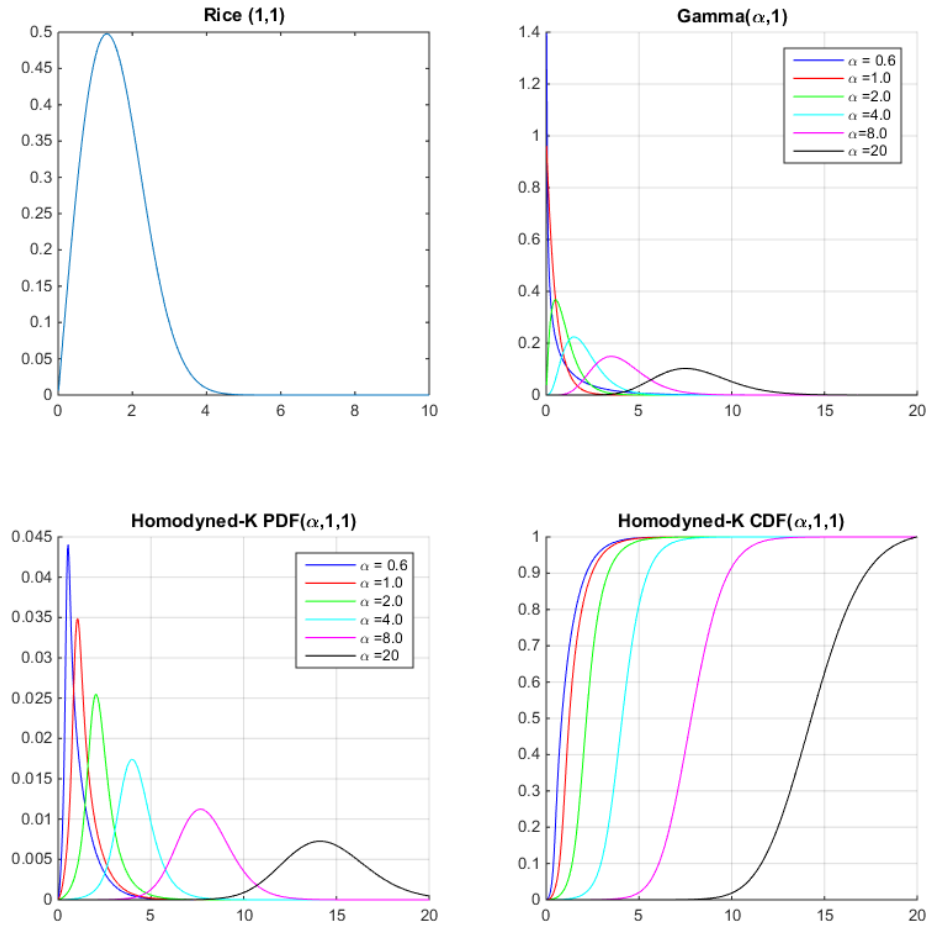


FIGURE 2.7: Homodyned K-distributions resulting of compounding a modulated Rice ($\nu = 1.0$) with different effective scatter densities ($0.5 \leq \alpha \leq 20$) modulating Gamma distribution.

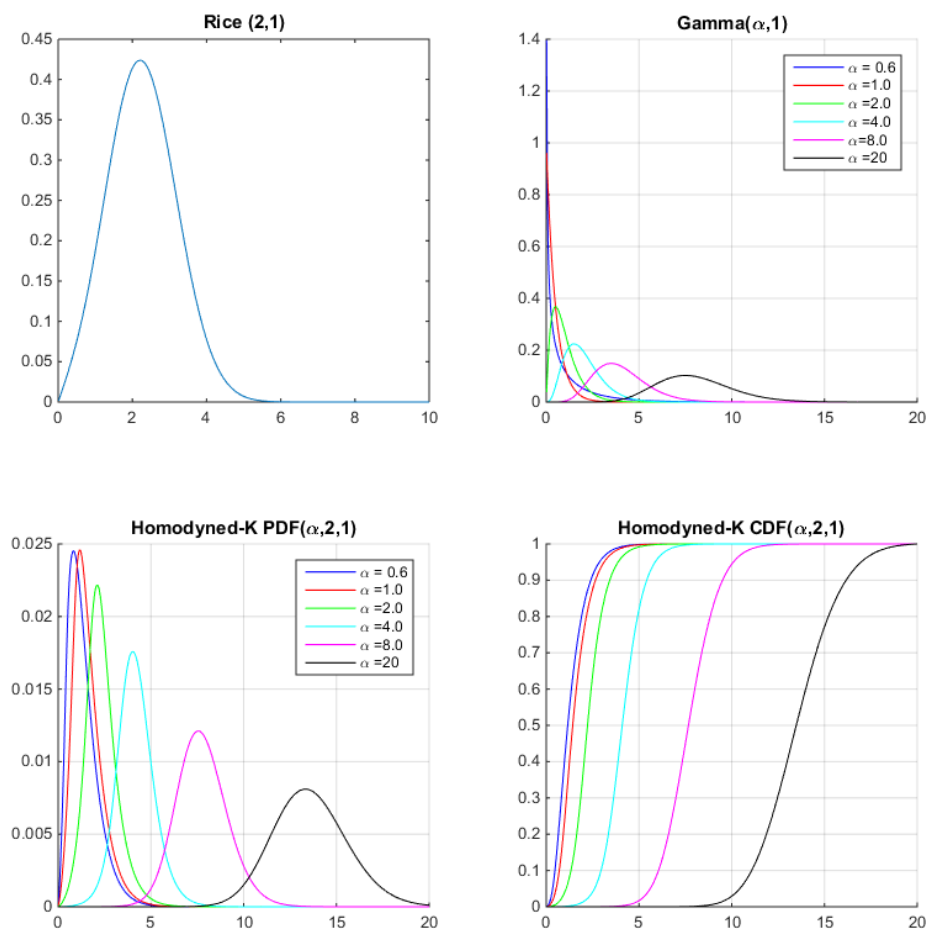


FIGURE 2.8: Homodyned K-distributions resulting of compounding a modulated Rice ($\nu = 2.0$) with different effective scatter densities ($0.5 \leq \alpha \leq 20$) modulating Gamma distribution.

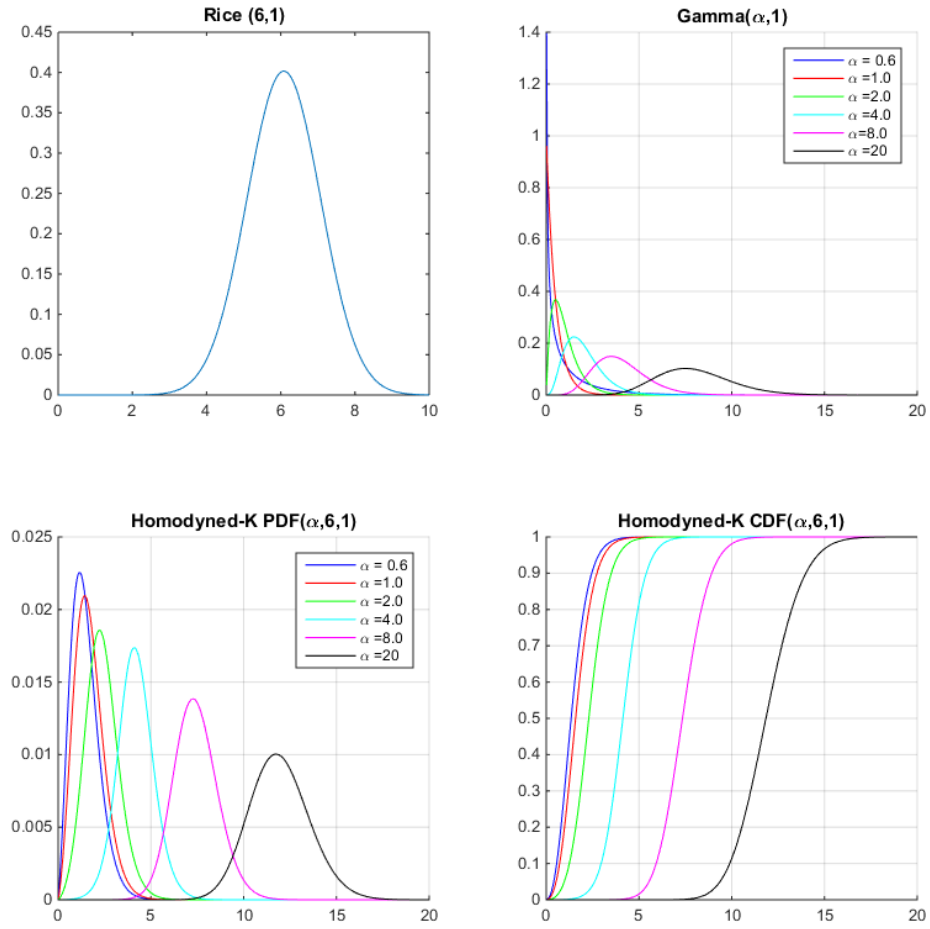


FIGURE 2.9: Homodyned K-distributions resulting of compounding a modulated Rice ($\nu = 6.0$) with different effective scatter densities ($0.5 \leq \alpha \leq 20$) modulating Gamma distribution.

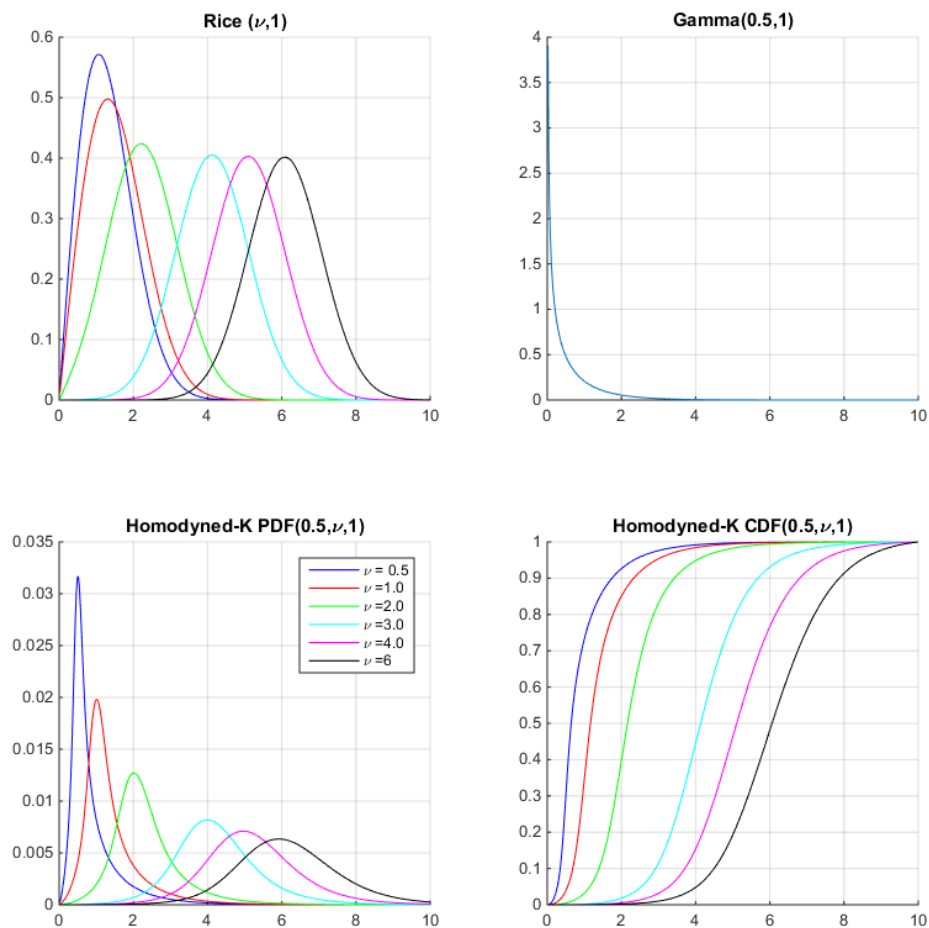


FIGURE 2.10: Homodyned K-distributions resulting of compounding different modulated Rice ($0.5 \leq \nu \leq 20$) distributions with very low effective scatter density ($\alpha = 0.5$) modulating Gamma distribution.

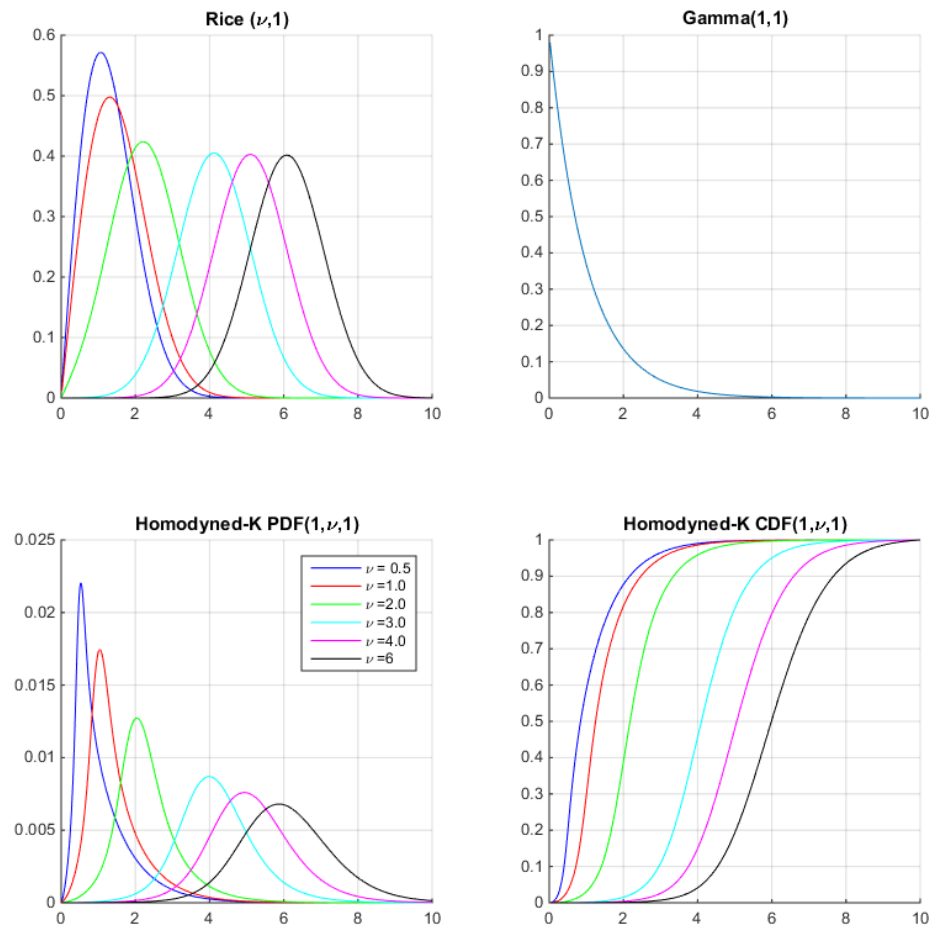


FIGURE 2.11: Homodyned K-distributions resulting of compounding different modulated Rice ($0.5 \leq \nu \leq 20$) distributions with low effective scatter density ($\alpha = 1.0$) modulating Gamma distribution.

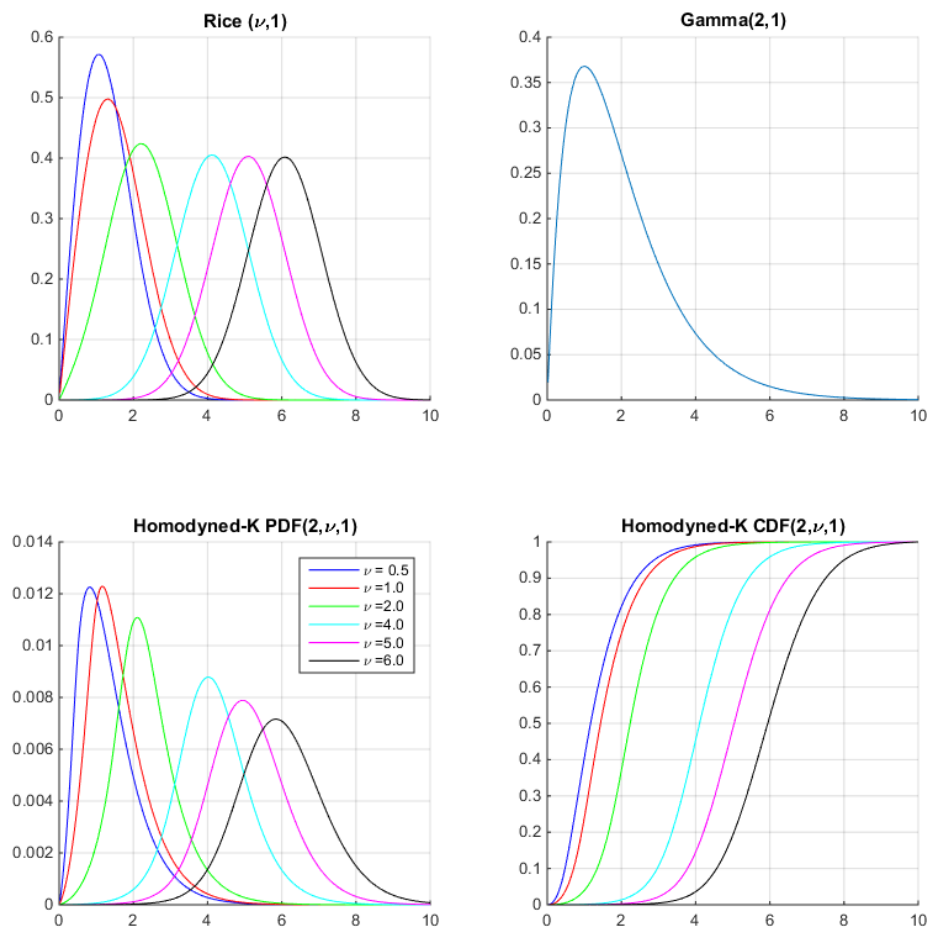


FIGURE 2.12: Homodyned K-distributions resulting of compounding different modulated Rice ($0.5 \leq \nu \leq 20$) distributions with $\alpha = 2.0$ effective scatter density modulating Gamma distribution.

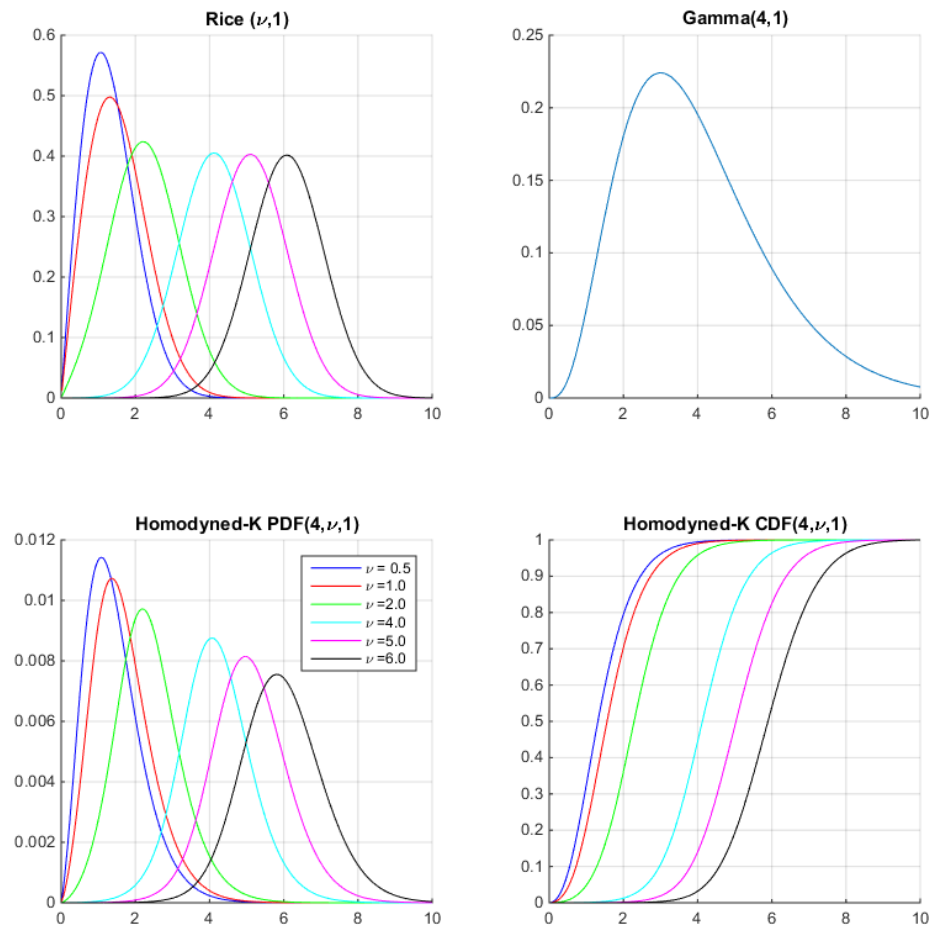


FIGURE 2.13: Homodyned K-distributions resulting of compounding different modulated Rice ($0.5 \leq \nu \leq 20$) distributions with $\alpha = 4.0$ effective scatter density modulating Gamma distribution.

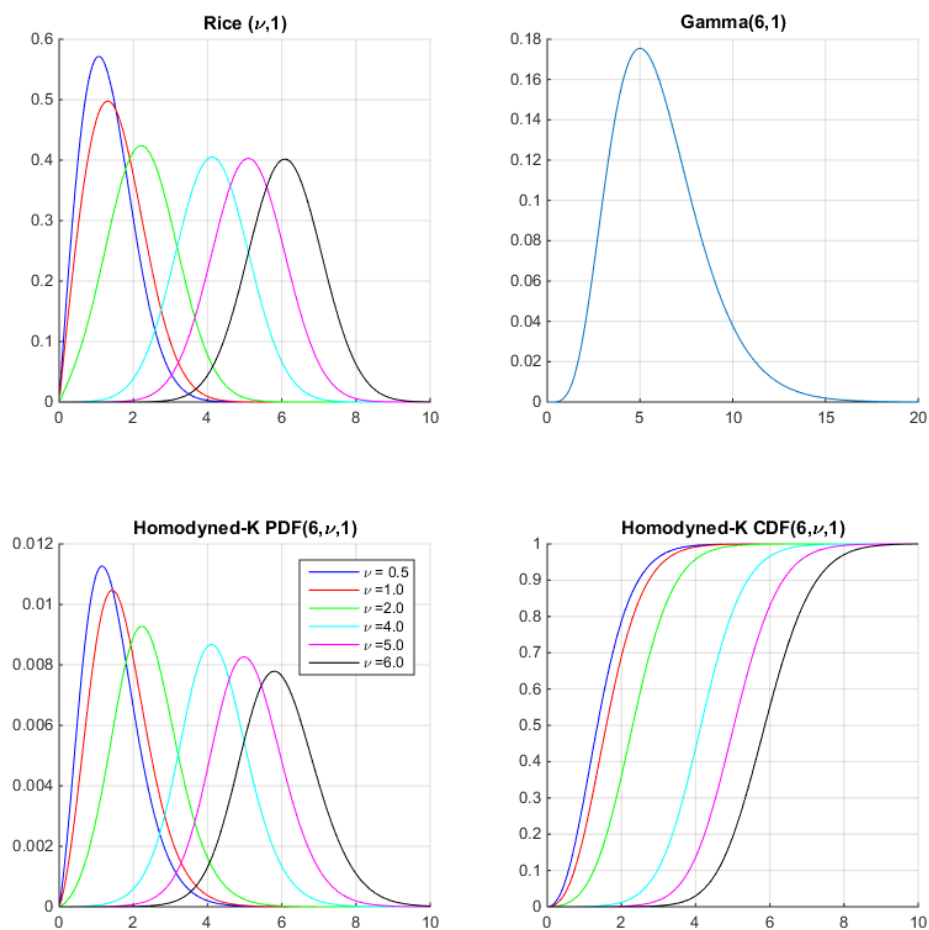


FIGURE 2.14: Homodyned K-distributions resulting of compounding different modulated Rice ($0.5 \leq \nu \leq 20$) distributions with $\alpha = 6.0$ effective scatter density modulating Gamma distribution.

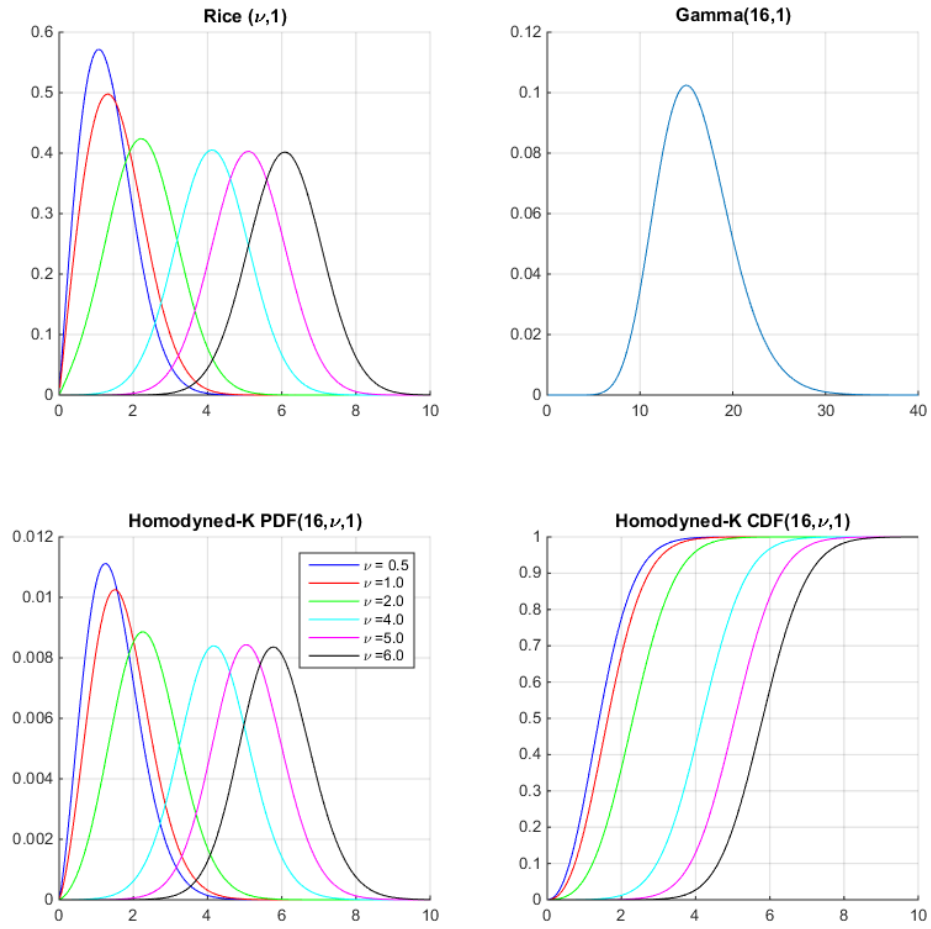


FIGURE 2.15: Homodyned K-distributions resulting of compounding different modulated Rice ($0.5 \leq \nu \leq 20$) distributions with $\alpha = 16.0$ effective scatter density modulating Gamma distribution.

Homodyned-K distributions are depicted in figures 2.5 to 2.8, for different parameters condition.

2.2.2 Generalized K-Distribution

From Jakeman and Tough, 1987 (equations 4.10 to 4.12) and used in ultrasound by Eltoft, 2006c the compound representation of the generalized K-distribution is

$$P(X|\sigma^2, \nu, \alpha) = \int_0^\infty P_r(X|\nu w, \sigma^2 w) P_\gamma(w|\alpha, 1) dw, \quad \alpha, \sigma, X > 0; \quad \nu \geq 0 \quad (2.16)$$

where P_r denotes the Rice distribution and P_γ is the Gamma Distribution with mean and variance equal to α . Thus, as opposed to the Homodyned K-distribution, both the coherent signal component and the diffuse signal power of the Rice distribution are modulated by a gamma distribution in Generalized-K distribution compound representation.

2.2.3 Rician Inverse Gaussian Distribution (RiIG)

From Eltoft, 2006b the compound representation of the Rician Inverse Gaussian Distribution (RiIG) is

$$P_{RiIG}(X|\nu\sigma^2, \alpha) = \int_0^\infty P_r(X|\nu w, \sigma^2 w) IG(w|\sqrt{\lambda}, \lambda) dw, \quad \lambda, \sigma, X > 0; \quad \nu \geq 0 \quad (2.17)$$

where P_r denotes the Rice distribution and $IG(w|\sqrt{\lambda}, \lambda)$ is Inverse Gaussian Distribution, with mean μ and shape parameter equal to λ :

$$IG(w|\sqrt{\lambda}, \lambda) = \sqrt{\frac{\lambda}{2\pi w^3}} \exp\left(-\frac{\lambda(w - \mu)^2}{2\mu^2 w}\right) \quad (2.18)$$

The mean and variance of $IG(w|\sqrt{\lambda}, \lambda)$ are both equal to $\sqrt{\lambda}$. So, in the compound representation context, $\sqrt{\lambda}$ plays the same role as α in the Gamma Distribution with mean and variance equal to α , however a parameter physical interpretation is not as straight forward.

2.3 Multiplicative model approach

The multiplicative model has been widely used in the modeling, processing, and analysis of synthetic aperture radar images. This model states that, under most conditions (Tur, Chin, and Goodman, 1982), the return results from the product between the speckle noise and the terrain backscatter. Several distributions could be used for the backscatter, aiming at the modeling of different types of classes and their characteristic degrees of homogeneity. For instance, for some sensor parameters (wavelength, angle of incidence, polarization, etc.), pasture is more homogeneous than forest, which, in turn, is more homogeneous than urban areas. Most distributions for the (amplitude) backscatter do not yield to closed-form distributions for the return, being a constant, a square root of Gamma and a square root of generalized inverse Gaussian distribution

important exceptions.

Common distributional hypothesis for one-look return data and homogeneous targets are the Exponential and Rayleigh distributions, for intensity and amplitude detections, respectively. When the observed region cannot be assumed as homogeneous, other distributions are considered; among these, the K-distributions have received a great deal of attention in the literature, and in chapter 3 a generalization will be exposed in detail.

2.3.1 Generalized K-Distribution

From Eltoft, 2006c a normal variance-mean mixture model, was presented. An equivalent 1-D Gamma variance-mean mixture model is in its most general form expressed as:

$$Y = (m + bZ + Z^{1/2}X) \quad m, b \geq 0 \quad (2.19)$$

$Z^{1/2}$ is a square root gamma distribution (i.e. Nakagami distribution) with parameters (1,1) equivalent to a Rayleigh distribution; and X is an independent square root gamma distribution with parameters $(\alpha, \gamma > 0)$. This mixture model has as particular cases:

- Rayleigh distribution: $m, b=0, \alpha \rightarrow \infty$.
- K- Distribution: $m, b=0$.
- Rice Distribution: $b=0, \alpha \rightarrow \infty$.
- Homodyned-k distribution: $b=0$.
- Generalized-k distribution: $m=0$.

Chapter 3

A general model for ultrasound B-scan images

3.1 Statistical models for SAR data

Among the frameworks for Synthetic Aperture Radar (SAR) image modelling and analysis, the multiplicative model is very accurate and successful. It is based on the assumption that the observed random field is the result of the product of two independent and unobserved random fields: X and Y . The random field X models the terrain backscatter and, thus, depends only on the type of area to which each pixel belongs to. The random field Y takes into account that SAR images are the result of a coherent imaging system that produces the speckle noise, and that they are generated by performing an average over statistically independent images (looks) in different bands or polarizations, in order to reduce the speckle noise effect. There are various ways of modelling the random field X ; while, with the usual Rayleigh distribution for the amplitude speckle, resulted in a different distribution families for the return. The distribution parameters depend on the reflective surface (ground in SAR data), and the number of looks. The advantage of the multiplicative model is that it can model extremely heterogeneous areas like cities, as well as moderately heterogeneous areas like forests and homogeneous areas like pastures. In this chapter, the successful multiplicative model will be adapted from SAR to ultrasound images. Then, the adapted distribution will be Log-compressed and the resulting model parameters will be estimated.

3.1.1 Speckle Noise Model

The speckle noise model, proposed by Arsenault Arsenault and April, 1976, is deduced from the coherent imaging mechanism of a SAR system. Under the ideal circumstances the imaged scene has a constant Radar Cross Section (RCS) (i.e., speckle is fully developed and homogeneous surfaces appear as stationary fields). The deducing process based on the coherent imaging mechanism begins with the six reasonable hypotheses as follows (Oliver and Quegan, 2004; Moser, Zerubia, and Serpico, 2006; Kuruoglu and Zerubia, 2004; Goodman, 1976):

- Each resolution cell contains sufficient scatterers.
- The echoes of these scatterers are independently identically distributed;
- The amplitude and phase of the echo of each scatterer are statistically independent random variables.
- The phase of the echo of each scatterer is uniformly distributed in the closed interval $[-\pi, \pi]$.

- Inside a resolution cell, there are no dominant scatterers.
- The size of a resolution cell is large enough, compared with the size of a scatterer.

Secondly, with the six hypotheses mentioned above and the central limit theorem, it can be proven that the energy of each resolution cell has a negative exponential distribution with the mean value equal to the real RCS value of the resolution cell. Finally, according to the hypothesis of constant RCS background, each resolution cell can be considered as a stochastic process, with the ergodic property (i.e., each resolution cell is statistically independent). Therefore, the whole image has a distribution identical to that of a single resolution cell.

3.1.2 The Multiplicative Model

Motivated by the speckle model, Ward Ward, 1981 proposed the Multiplicative Model of SAR images. The Multiplicative Model combines an underlying RCS component σ with an uncorrelated multiplicative speckle component n , so the observed intensity I in a SAR image can be expressed as the product (Tur, Chin, and Goodman, 1982, Xie, Pierce, and Ulaby, 2002):

$$I = \sigma \cdot n \quad (3.1)$$

The speckle model is taken as the special example of the Multiplicative Model with a constant RCS (σ). Because the Multiplicative Model is correlated with the underlying terrain RCS (σ), it is usually applied to the intensity data (energy or the square of amplitude). That is, I in Equation 3.1 represents the observed value of the intensity. The Multiplicative Model simplifies the analysis of the statistical model of SAR images. So it is widely used to develop models which take the RCS fluctuations into consideration, where $P(\sigma)$ represents the RCS component distribution and $P(I|\sigma)$ is correlated with the distribution of speckle component.

Since the speckle component has a determinate statistical distribution, only the RCS fluctuation component need to be considered when developing the statistical models of SAR images, as a result according 3.1, the PDF of the observed intensity is given by:

$$P(I) = \int P(\sigma) \cdot P(I | \sigma) d\sigma \quad (3.2)$$

The RCS of a homogeneous region (e.g., the grassland region) in either low-resolution or high-resolution SAR images can be expected to correspond to a constant. Actually, most scenes contain in-homogeneous regions with RCS fluctuations (Oliver and Quegan, 2004; Kuruoglu and Zerubia, 2004; Moser, Zerubia, and Serpico, 2006). According to Jakeman and Pusey, 1976 investigations, when the number of scatterers in a resolution cell becomes a random variable due to fading phenomenon and the population of scatterers to be controlled by a birth-death-migration process, it should have a Poisson distribution Oliver and Quegan, 2004 and the mean of the Poisson distribution in each resolution cell (i.e., the expected number of scatterers) itself is also a random variable. If the mean is Gamma distributed, the corresponding intensity data should have a K distribution. Further research indicates that K distribution can be viewed as the combination of two split parts according to Equation 3.2 in the framework of the Multiplicative Model (Oliver and Quegan, 2004):

- the speckle component satisfying the central limit theorem.

- the Gamma distributed intensity RCS fluctuations.

The K distribution is deduced with the assumption that the underlying intensity RCS fluctuations have a Gamma distribution in a heterogeneous region. The Gamma distribution can well describe the characteristics of the RCS fluctuations of a heterogeneous terrain in high-resolution SAR images. The deduced K distribution itself has the multiplicative fading statistical characteristics and usually provides a good fit to the heterogeneous terrain. Therefore, the K distribution has become one of the most widely used and the most famous statistical models. Furthermore, according to the deducing process of the K distribution, the homogeneous region with a constant RCS can also be described as a special case of the K distribution (Oliver and Quegan, 2004). However, K-distributions cannot meet the demand for the statistical modeling of complex scenes in high-resolution images. The complexity of the high-resolution scenes mainly lies in two aspects (Kuruoglu and Zerubia, 2004):

- The terrain of the scene is usually extremely heterogeneous, such as the urban region containing many buildings, which results in the severe long-tailed part of the image histogram.
- There exist two or more heterogeneous components in a certain scene, such as a combination of woodlands and grasslands, etc.

To solve these problems, Frery et al., 1997 deduced a new statistical model, the GA model, based on the product model assuming a Gamma distribution for the speckle component of multi-look SAR images and a generalized inverse Gaussian (GIG) law for the signal component. It was Frery who first proposed to divide a region as homogeneous, commonly heterogeneous or extremely heterogeneous according to its homogeneous degree when deducing the GA model. The K and G0 (also called B distribution) distributions are two special forms of the G model. The former is appropriate for the heterogeneous region and the latter is appropriate for the extremely heterogeneous region. The G0 distribution can be converted into the Beta-Prime distribution under the single-look condition. Although the G0 distribution is a specific example of the G model, it has a more compact form in comparison with the G model and consequently has a simple parameter estimation method. The relationship between the G0 distribution and the K distribution cannot be deduced theoretically, but has been evaluated via Montecarlo Simulation Methods (Mejail et al., 2001). The parameters of the G0 distribution are sensitive to the homogeneous degree of a region, which makes the G0 model appropriate for modeling either heterogeneous or extremely heterogeneous region. Moreover, moments method can be easily and successfully applied to parameter estimation of the G0 distribution; and the Log-Compressed G0 distribution, namely HG0 distribution, has an analytical expression. Also, Frery et al., 1997 and Muller and Pac, 1999 carried out experiments on many SAR images of different kinds of terrain with various band, polarization, resolution and look numbers, such as different urban areas, homogeneous and heterogeneous regions, etc. Their results testified the good characteristics of the G0 distribution. In next sections the GA and GA0 models will be presented and adapted to ultrasound B-scan images modeling.

3.2 G_A distributions for SAR images

Equation 3.1 can be rewritten in equivalent form:

$$I = X \cdot Y \quad (3.3)$$

Equation 3.3 states that the intensity observed value I is the outcome of a random variable defined by the product of two independent random variables: X modeling the terrain backscatter, and Y modeling the speckle noise. In a general situation when independent bands or polarizations images are averaged to reduce speckle noise, it is necessary to take into account the number of looks n ; then multi-look intensity speckle appears by taking an average over n independent samples, leading, thus to a Gamma distribution denoted as $Y_I \sim \Gamma(n, n)$ and characterized by the density:

$$f_{Y_I}(y) = \frac{n^n}{\Gamma(n)} y^{n-1} \exp(-ny), \quad y, n > 0$$

Multi-look amplitude speckle results from taking the square root of the multi-look intensity speckle and, therefore, has a square root of Gamma distribution, denoted as $Y_A \sim \Gamma^{1/2}(n, n)$ and characterized by the density:

$$f_{Y_A}(y) = \frac{2n^n}{\Gamma(n)} y^{2n-1} \exp(-ny^2), \quad y, n > 0$$

Though the number of looks should, in principle, be an integer, seldom this is the case when this quantity is estimated from real data due to, among other reasons, the fact that the mean is taken over correlated observations. Therefore, the equivalent number of looks must be estimated (see for example Anfinssen, Doulgeris, and Eltoft, 2008).

Frery et al., 1997 proposed modeling amplitude backscatter with the Square Root Generalized Inverse Gaussian Law, i.e. $X_A \sim N^{-1/2}(\alpha, \gamma, \lambda)$, with density given by

$$f_{X_A}(X) = \frac{(\lambda/\gamma)^{\alpha/2}}{K_\alpha(2\sqrt{\lambda\gamma})} x^{2\alpha-1} \exp\left(-\frac{\gamma}{x^2} - \lambda x^2\right)$$

where K_α denotes the modified Bessel function of the second kind and order α . The parameters space is given by

$$\begin{cases} \gamma > 0, & \lambda \geq 0, & \text{if } \alpha < 0 \\ \gamma > 0, & \lambda > 0, & \text{if } \alpha = 0 \\ \gamma \geq 0, & \lambda > 0, & \text{if } \alpha > 0 \end{cases} \quad (3.4)$$

The amplitude return, $G_A(\alpha, \gamma, \lambda)$, that arises from the product of $X_A \cdot Y_A = Z_A \sim G_A(\alpha, \gamma, \lambda)$, where $X_A \sim N^{-1/2}(\alpha, \gamma, \lambda)$ and $Y_A \sim \Gamma^{1/2}(n, n)$; is characterized by the density

$$f_{Z_A}(x) = \frac{2n^n(\lambda/\gamma)^{\alpha/2}}{\Gamma(n)K_\alpha(2\sqrt{\lambda\gamma})} x^{2n-1} \left(\frac{\gamma + nx^2}{\lambda}\right)^{(\alpha-n)/2} K_{\alpha-n}\left(2\sqrt{\lambda(\gamma + nx^2)}\right) \quad (3.5)$$

and parameters space given in equation 3.4.

From $N^{-1/2}$, the Square Root Gamma Distribution ($\Gamma^{1/2}(x, \alpha, \lambda)$) arises by letting $\gamma \rightarrow 0$ while $\alpha, \lambda > 0$. This distribution is characterized by the density

$$f_{X_A}(x) = \frac{2\lambda^\alpha}{\Gamma(\alpha)} x^{2\alpha-1} \exp(-\lambda x^2), \quad \alpha, \lambda, x > 0$$

Also, the Reciprocal Square Root Gamma Distribution ($\Gamma^{-1/2}(x, \alpha, \gamma)$) arises by letting $\lambda \rightarrow 0$ while $\alpha, \gamma > 0$. This distribution is characterized by the density

$$f_{X_A}(x) = \frac{2}{\gamma^\alpha \Gamma(\alpha)} x^{-(2\alpha+1)} \exp(-\gamma/x^2), \quad \alpha, \lambda, x > 0$$

It should be noticed that, if $X \sim \Gamma^{1/2}(x, \eta, \gamma)$, then $Z = 1/X \sim \Gamma^{-1/2}(z, \alpha, \gamma^{-1})$ with $\alpha = \eta$.

Therefore, $\Gamma^{1/2}(\alpha_m, \lambda_m)$ and $\Gamma^{-1/2}(\alpha_m, \gamma_m)$ distributions are particular cases of $N^{-1/2}(\alpha, \gamma, \lambda)$ distribution; and also, using characteristic functions it can be proved that a sequence of random variables obeying $\Gamma^{1/2}(\alpha_m, \lambda_m)$ distributions converges in probability to the constant $\beta_1^{1/2}$, if $\alpha_m, \lambda_m \rightarrow \infty$ such that $\alpha/\lambda \rightarrow \beta_1$ when $m \rightarrow \infty$.

As a result, a sequence of random variables obeying $\Gamma^{-1/2}(\alpha_m, \gamma_m)$ distributions converges in probability to the constant $\beta_2^{-1/2}$, if $\alpha_m, \gamma_m \rightarrow \infty$ such that $\alpha/\gamma \rightarrow \beta_2$ when $m \rightarrow \infty$.

In this manner, constant amplitude backscatter, used to model homogeneous areas, arises in two situations and are particular cases of the square root of generalized Inverse Gaussian Distribution (GiG).

3.2.1 K_A -distribution for SAR image

The K-distribution amplitude return, $K_A(\alpha, \lambda)$, that arises from the product of $X_A \cdot Y_A = Z_A \sim K_A(\alpha, \lambda)$, where $X_A \sim \Gamma^{1/2}(\alpha, \lambda)$ and $Y_A \sim \Gamma^{1/2}(n, n)$; is characterized by the density

$$f_{Z_A}(x) = \frac{4\lambda n x}{\Gamma(n)\Gamma(\alpha)} (\lambda n x^2)^{(\alpha+n)/2-1} K_{\alpha-n}(2x\sqrt{\lambda n}) \quad (3.6)$$

3.2.2 GA0 distribution for SAR image

GA0 distribution, $G_{A0}(\alpha, \gamma)$, arises from the product of $X_A \cdot Y_A = Z_A \sim G_{A0}(\alpha, \gamma)$, where $X_A \sim \Gamma^{-1/2}(\alpha, \gamma)$ and $Y_A \sim \Gamma^{1/2}(n, n)$; is characterized by the density

$$f_{Z_A}(x) = \frac{2n^n \Gamma(\alpha + n) \gamma^\alpha x^{2n-1}}{\Gamma(n) \Gamma(\alpha) (\gamma + nx^2)^{\alpha+n}} \quad (3.7)$$

3.3 G0 distribution for ultrasound image processing

Acoustical waves are longitudinal, therefore cannot be polarized. Also, ultrasound images are single look i.e. effective number of looks is just $n=1$; then, amplitude return $A=XY$ is distributed as (Springer, 1979):

$$G0(A) = \int_{-\infty}^{+\infty} (1/X) f_x(X) f_y(A/X) dX$$

With X distributed as:

$$f_x(x) = \Gamma^{1/2}(x, 1, 1) = 2x \exp(-x^2), \quad x > 0$$

and Y with density

$$f_y(y) = \Gamma^{-1/2}(y, \alpha, \gamma) = \frac{2\gamma^\alpha \exp(-\gamma/y^2)}{\Gamma(\alpha)y^{2\alpha+1}}, \quad y, \alpha, \gamma > 0$$

Therefore $G0(A)$ can be written as:

$$G0(A) = \int_{-\infty}^{+\infty} (1/x) (2x \exp(-x^2)) \left(\frac{2\gamma^\alpha \exp(-\gamma x^2/A^2)}{\Gamma(\alpha)(A/x)^{2\alpha+1}} \right) dx$$

$$G0(A) = \frac{2 * 2 * \gamma^\alpha}{\Gamma(\alpha) A^{2\alpha+1}} \int_{-\infty}^{+\infty} (x^{2\alpha+1}) \exp(-x^2(1 + \frac{\gamma}{A^2})) dx$$

Replacing:

$$u = x^2(1 + \frac{\gamma}{A^2}), \quad x^2 = u \left(\frac{A^2}{A^2 + \gamma} \right), \quad 2x dx = du \left(\frac{A^2}{A^2 + \gamma} \right),$$

We obtain:

$$G0(A) = \frac{2\gamma^\alpha}{\left(1 + \frac{\gamma}{A^2}\right)^{\alpha+1} \Gamma(\alpha) A^{2\alpha+1}}$$

Using the Gamma Function definition:

$$\int_0^{+\infty} u^\alpha \exp(-u) du = \Gamma(\alpha + 1) = \alpha \Gamma(\alpha)$$

$G0$ distribution is obtained:

$$G0(A, \alpha, \gamma) = \frac{2\alpha(A/\gamma)}{[1 + (A^2/\gamma)]^{\alpha+1}}, \quad \alpha, \gamma, A > 0 \quad (3.8)$$

Here, the random variable A (detected amplitude) results from the product of the random variables x_n and y_e . Subindex n represents the speckle noise, while e is the backscatter distribution.

3.4 HG0 distribution for Log-compressed B-scan images

Clinical ultrasound imaging systems employ nonlinear signal processing to reduce the dynamic range of the input echo signal to match the smaller dynamic range of the display device and to emphasize objects with weak backscatter. Typically, the input image could have dynamic ranges of the order of 50-70 dB whereas a display device would have dynamic range of the order of 20-30 dB. This reduction in dynamic range is normally achieved through a logarithmic compression which selectively compresses large input signals.

This kind of nonlinear compression totally changes the statistics of the input envelope signal. A closed form expression for the density function of the log transformed distributed data usually is hard to derive. However, it is possible to obtain the density function of log-compressed G0 density function, as will be shown in this section.

3.4.1 Logarithmic Compression Model

The logarithmic compression transfer function can be written as

$$X = D \cdot \ln(A) + G \quad (3.9)$$

where A is the input to the compression block and X is the output of the compression block. D is a parameter of the compressor which represents the dynamic range of input, and G is the linear gain of the compressor. Here it is assumed that the input is nonzero.

The linear gain parameter, G , does not affect the statistics of the output signal because it just changes the mean of the distribution function. But the dynamic range parameter, D , scales the output signal and is thus important to estimate if one has to invert this logarithmic transfer function.

If the minimum and maximum input values A_{min} and A_{max} , are mapped to minimum and maximum output values X_{min} and X_{max} by this logarithmic compression, then the relationship between them can be written as

$$\begin{aligned} X_{max} &= D \cdot \ln(A_{max}) + G \\ X_{min} &= D \cdot \ln(A_{min}) + G \end{aligned}$$

Therefore

$$\begin{aligned} D &= (X_{max} - X_{min}) / \ln(A_{max}/A_{min}) \\ G &= X_{min} - D \cdot \ln(A_{min}) \end{aligned}$$

These are the optimum logarithmic amplifier parameters, D and G , when the input-output dynamic ranges are known. For gray level displays, $X_{min} = 0$ and $X_{max} = 255$; and a typical input dynamic range, $A_{min} = 0.1mV$, $A_{max} = 10V$; result in optimum logarithmic amplifier parameters of $D = 22.15$ and $G = 204$.

3.4.2 A New Statistical Model of Log-compressed B-scan images

When the non-linear transform $z = D \cdot \ln(A) + G$ is applied to Amplitude Distribution f_A , it is transformed as:

$$f_z(z) = (1/D) \exp\left(\frac{z-G}{D}\right) f_A\left(\exp\left(\frac{z-G}{D}\right)\right)$$

The new random variable z when f_A is a Go distribution, is distributed as:

$$f_z(z) = \frac{\alpha \left(\frac{\exp\left(\frac{z-G}{D/2}\right)}{\gamma} \right)}{\frac{D}{2} \left(1 + \frac{\exp\left(\frac{z-G}{D/2}\right)}{\gamma} \right)^{\alpha+1}}$$

Replacing $d = D/2$ and $\exp(-g) = \exp(-\frac{G}{D/2})/\gamma$, HG0 distribution is obtained:

$$H_{Go}(z) = \frac{\alpha \exp((z-g)/d)}{d (1 + \exp((z-g)/d))^{\alpha+1}}, \quad \alpha, g, d > 0 \quad (3.10)$$

It is straight forward to show that the HG0 cumulative distribution function can be expressed as:

$$F_{HG0}(z|\alpha, d, g) = 1 - \frac{1}{[1 + \exp((z-g)/d)]^{\alpha}}, \quad \alpha, d, g > 0 \quad (3.11)$$

Although in principle $-\infty < z < \infty$ in practice the logarithmic amplifier G is adjusted for $z \geq 0$, i.e. noise levels under A_{min} are mapped into 0.

3.5 Moments generating function method for HG0 parameter estimation

Moments generating function is defined as the expected value of $\exp(tX)$, if $X \sim P_X(x)$

$$m_X(t) = \langle \exp(tX) \rangle = \int_S \exp(tX) P_X(x) dx$$

integration over sample space S ; for HG0 distribution, it means:

$$m_X(t) = \langle \exp(tz) \rangle = \frac{\alpha}{d} \int_{-\infty}^{\infty} \frac{\exp(tz) \exp(\frac{z-g}{d})}{[1 + \exp(\frac{z-g}{d})]^{\alpha+1}}$$

The moments generating function is (see Appendix A for details)

$$m_X(t) = \frac{\exp(gt)}{\Gamma(\alpha)} \Gamma(1 + td) \Gamma(\alpha - td) \quad (3.12)$$

and the normalized central moments can be expressed as:

$$\begin{cases} \mu_1 = g - d(\psi^{(0)}(1) - \psi^{(0)}(\alpha)) \\ \mu_2 = d^2(\psi^{(1)}(1) + \psi^{(1)}(\alpha)) \\ \mu_3 = d^3(\psi^{(2)}(1) - \psi^{(2)}(\alpha)) \end{cases} \quad (3.13)$$

where $\psi^{(n)}(x)$ is the poly-gamma function. From second and third normalized central moments, α is the solution of the non-linear equation

$$\frac{\mu_3}{\mu_2^{3/2}} = \mu_3 = \frac{(\psi^{(2)}(1) - \psi^{(2)}(\alpha))}{(\psi^{(1)}(1) + \psi^{(1)}(\alpha))^{3/2}} \quad (3.14)$$

Equation 3.14 can be solved with standard numerical methods. Parameters (d, g) , knowing α , can be expressed easily,

$$\begin{aligned} d &= (\psi^{(1)}(1) + \psi^{(1)}(\alpha))^{-1/2} \\ g &= d \cdot (\psi^{(0)}(1) - \psi^{(0)}(\alpha)) \end{aligned} \quad (3.15)$$

3.6 Maximum likelihood method for HG0 parameter estimation

From equation 3.10 the optimization function can be expressed as

$$L_{HG0} = \sum \ln(\alpha) + (z - g)/d - \ln(d) - (\alpha + 1) \ln(1 + \exp((z - g)/d)) \quad (3.16)$$

And the maximum conditions, are the partial derivatives of equation 3.16 with respect to the distribution parameters

$$\begin{cases} \frac{\partial L}{\partial \alpha} = 0 = 1 - \frac{\alpha}{N} \sum_i \ln[1 + \exp((z - g)/d)] \\ \frac{\partial L}{\partial g} = 0 = 1 + \frac{(\alpha + 1)}{N} \sum_i \frac{\exp((z - g)/d)}{(1 + \exp((z - g)/d))} \\ \frac{\partial L}{\partial d} = 0 = 1 - \frac{(\alpha + 1)}{d \cdot N} \sum_i \frac{d \cdot z \cdot \exp((z - g)/d)}{(1 + \exp((z - g)/d))} - \bar{z}/d \end{cases} \quad (3.17)$$

The non-linear system 3.17 can be solved with standard numerical methods. Note that this system does not involve Bessel functions, making this system less complex than those from K-distributions models. Also, it is apparent that the system describes the parameter condition of the ultrasound log-compressed B-scan images, therefore it is not necessary to invert the logarithmic amplifier equations, and even better, the logarithmic amplifier parameters are included as model parameters and can be estimated with both estimation methods: Methods Of Moments (MoM) described by the non-linear equations system 3.13 and the Maximum Likelihood Estimation (MLE) method described by equation system 3.17.

Chapter 4

Log-compressed data modeling

A Montecarlo study is performed with the distributions presented in Chapter 2. Log-compressed data are simulated and then they are modeled with the new model developed in Chapter 3. Finally, goodness of fit tests are performed for null hypothesis testing.

4.1 Montecarlo simulation of Log-compressed data

Rayleigh, Rice and K distributions are particular cases of the Homodyned-K and stable distributions, therefore this study will be focused in modeling Log-compressed Homodyned-K and stable distributed data.

4.2 Homodyned-K distributed data simulation

Simulating Homodyned-K distributed data can be done by using the compound representation (See chapter 2).

$$P(X|\sigma^2, \nu, \alpha) = \int_0^\infty P_r(X|\nu, \sigma^2 w) P_\gamma(w|\alpha, 1) dw, \quad \alpha, \sigma, X > 0; \quad \nu \geq 0 \quad (4.1)$$

Given parameters (ν, α) i.e. coherent component ν and effective scatterers density α , the algorithm to generate an length N - array A of Homodyned-K distributed data, can be expressed as:

- for $i=1$ to N
 - Generate σ number using Gamma distribution with parameters $(\alpha, 1)$
 - Generate $A(i)$ number using Rician distribution with parameters (ν, α)
- next i

4.3 Stable process simulation

Simulating stable distributions can be done with open source code (See appendix C). Then, the algorithm for stable process simulation can be stated:

- for i= 1 to N
 - Generate complex Z random variable with X real and Y imaginary stable variables with parameters $(\alpha, 1)$: $Z = X + iY$.
 - Generate random complex η with magnitude ν
 - Stable process is the magnitude of the complex $\eta + Z$.
- next i

4.4 Log Compression

Then, given Amax and Amin, maximum and minimum Homodyned-K and stable process distributed Amplitudes, the optimum logarithmic amplifier parameters are estimated using equation 4.2

$$\begin{aligned} D &= 255/\ln(A_{max}/A_{min}) \\ G &= -D \cdot \ln(A_{min}) \end{aligned} \tag{4.2}$$

Log-compressed data are the result of non-linear transform:

$$Z = D \cdot \log A + G$$

4.5 HG0 distribution for modeling Log-compressed data

Next, HG0 distribution parameters are estimated using MoM equation System 4.3:

$$\begin{aligned} \mu_3 &= \frac{(\psi^{(2)}(1) - \psi^{(2)}(\alpha))}{(\psi^{(1)}(1) + \psi^{(1)}(\alpha))^{3/2}} \\ d &= (\psi^{(1)}(1) + \psi^{(1)}(\alpha))^{-1/2} \\ g &= d \cdot (\psi^{(0)}(1) - \psi^{(0)}(\alpha)) \end{aligned} \tag{4.3}$$

With parameters (α, d, g) , generate HGO distributed data Z1 and finally perform a Two sample Kolmogorov-Smirnov test.

Two-sample Kolmogorov-Smirnov goodness-of-fit hypothesis test is used to determine if independent random samples, Z and Z1, are drawn from the same underlying continuous population. H indicates the result of the hypothesis test:

$H = 0 \Rightarrow$ Do not reject the null hypothesis at the 5% significance level.

$H = 1 \Rightarrow$ Reject the null hypothesis at the 5% significance level.

Let $S1(z)$ and $S2(z1)$ be the empirical distribution functions from the sample vectors Z and $Z1$, respectively, and $F1(x)$ and $F2(x)$ be the corresponding true (but unknown) population CDFs. The two-sample K-S test tests the null hypothesis that $F1(z) = F2(z1)$ for all z , against the alternative that they are unequal.

The decision to reject the null hypothesis occurs when the significance level equals or exceeds the P-value= 0.05.

4.6 Concluding Remarks

The Montecarlo experiments show that the HG0 distribution can describe data in all the situations covered with the Homodyned-K and stable distributions. The experiments also show a strong correlation between the α parameter, the effective scatterers density; and the ν parameter which account for the coherent component presence; due, probably, to the invariant characteristic of both parameters against scaling. In next chapter theoretical basis for this phenomenon will be given.

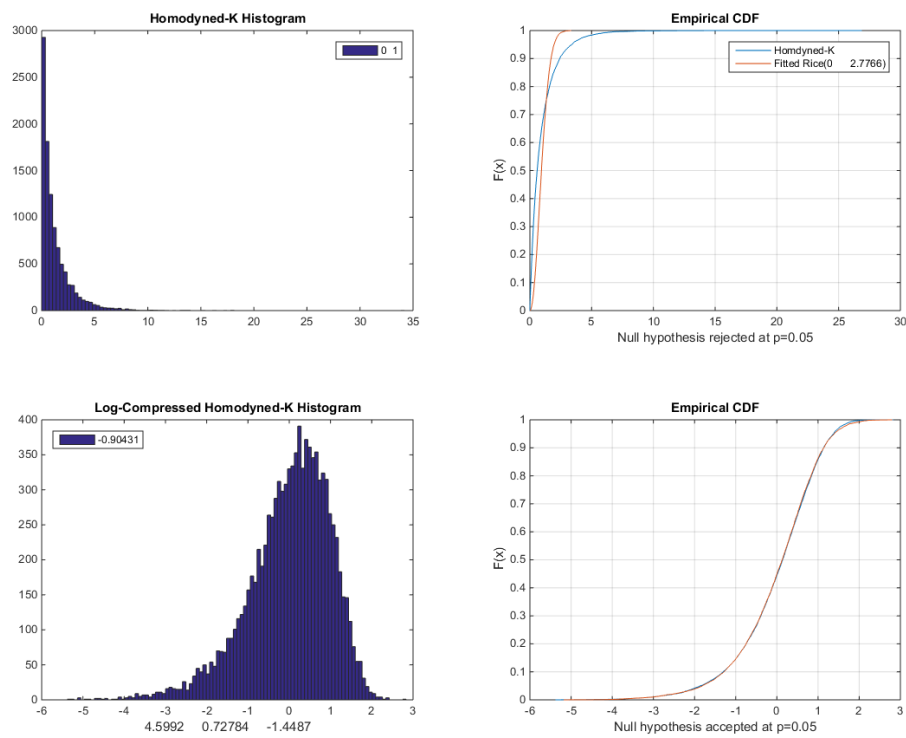


FIGURE 4.1: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Low effective density $\alpha = 1$ and no coherent component.

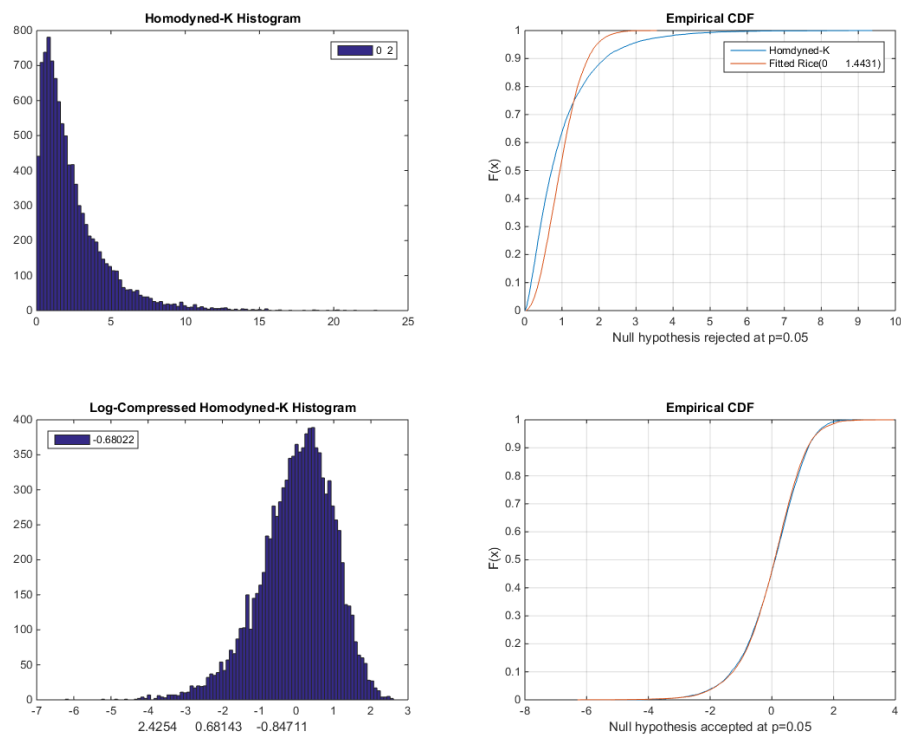


FIGURE 4.2: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Low effective density $\alpha = 2$ and no coherent component.

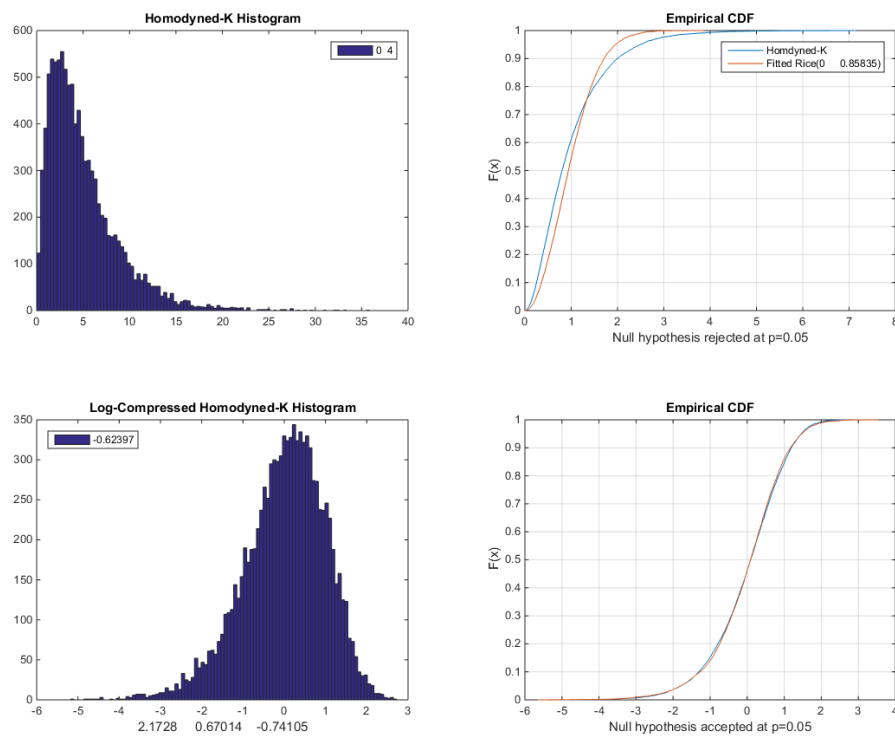


FIGURE 4.3: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Low effective density $\alpha = 4$ and no coherent component.

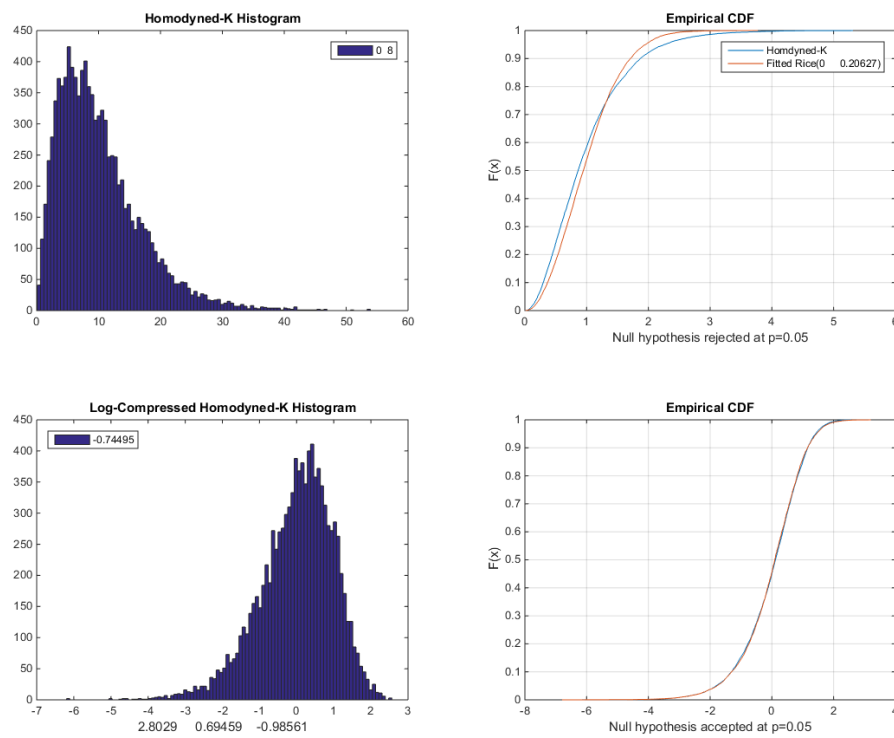


FIGURE 4.4: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Medium effective density $\alpha = 8$ and no coherent component.

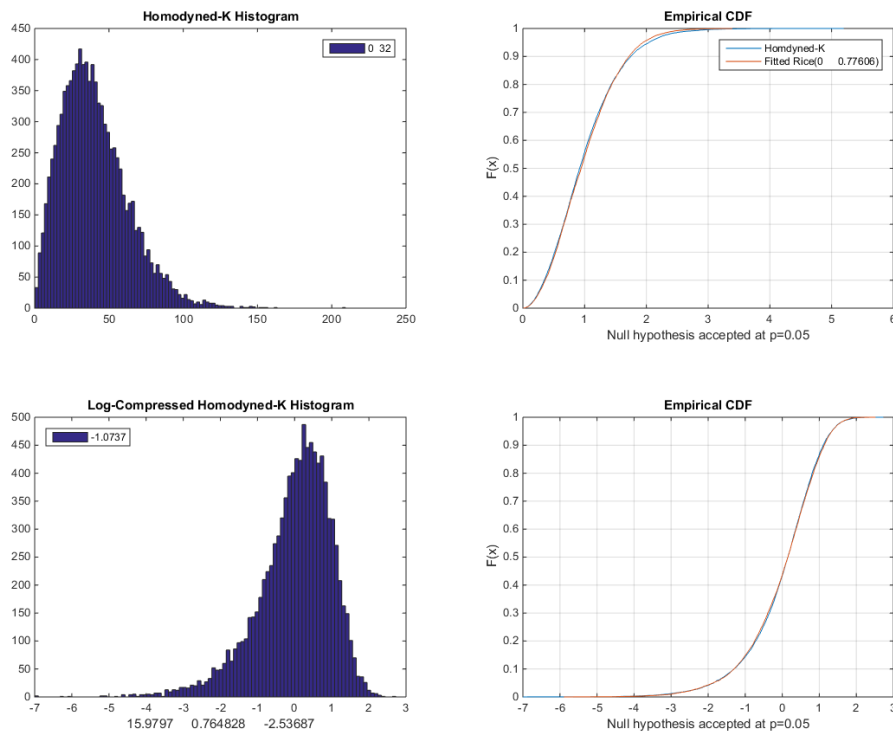


FIGURE 4.5: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. High effective density $\alpha = 32$ and no coherent component. Note the limiting case of $\alpha \rightarrow \infty$ is reached and the Rician distribution with no coherent component is fitted with Rayleigh distribution

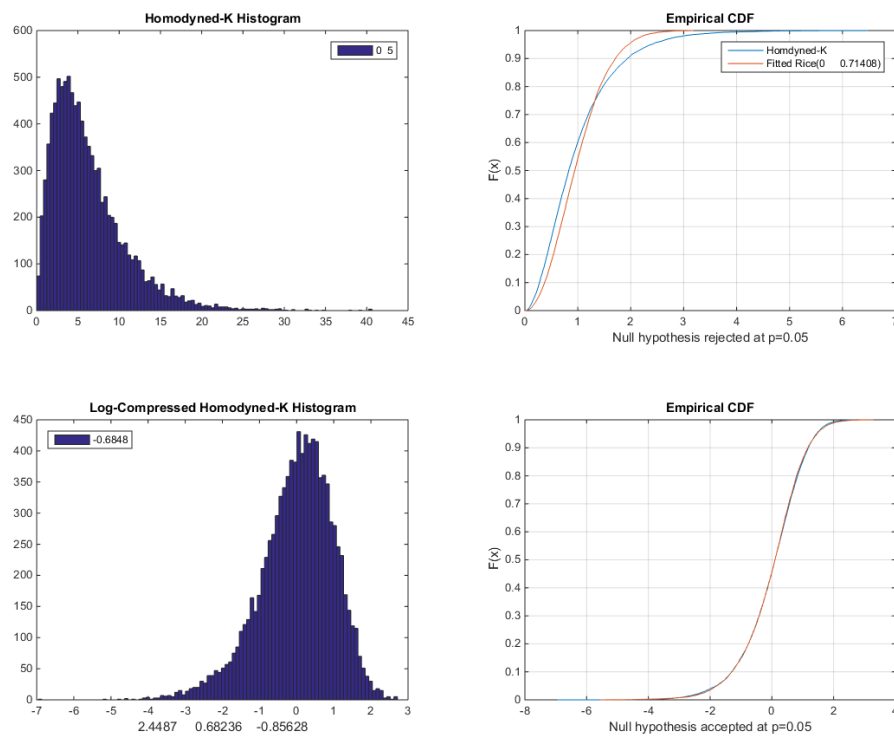


FIGURE 4.6: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Medium effective density $\alpha = 5$ and no coherent component.

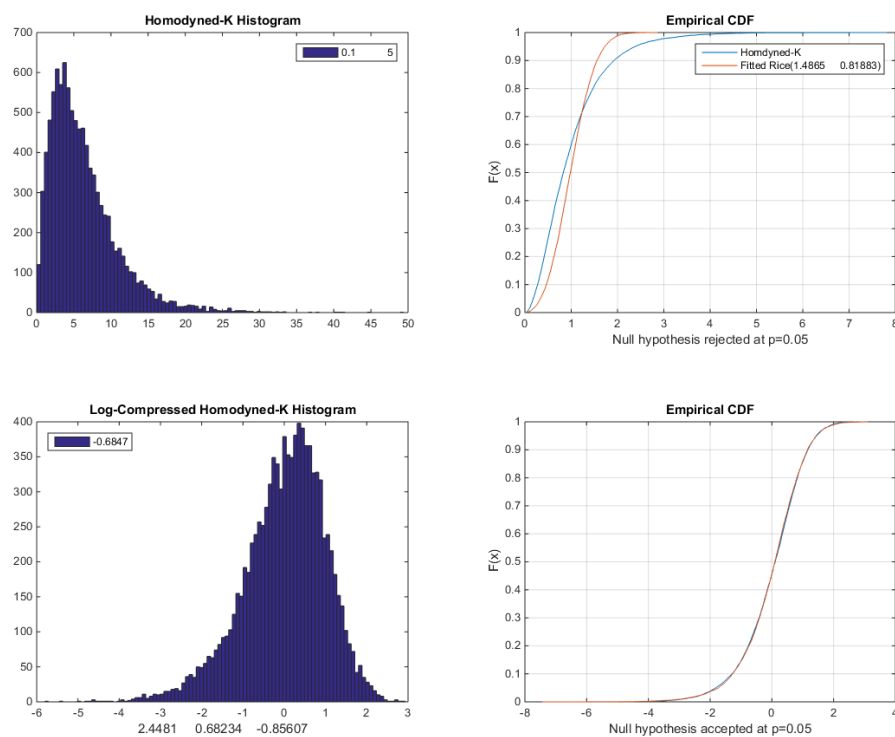


FIGURE 4.7: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Medium effective density $\alpha = 5$ and low coherent component $\nu = 0.1$

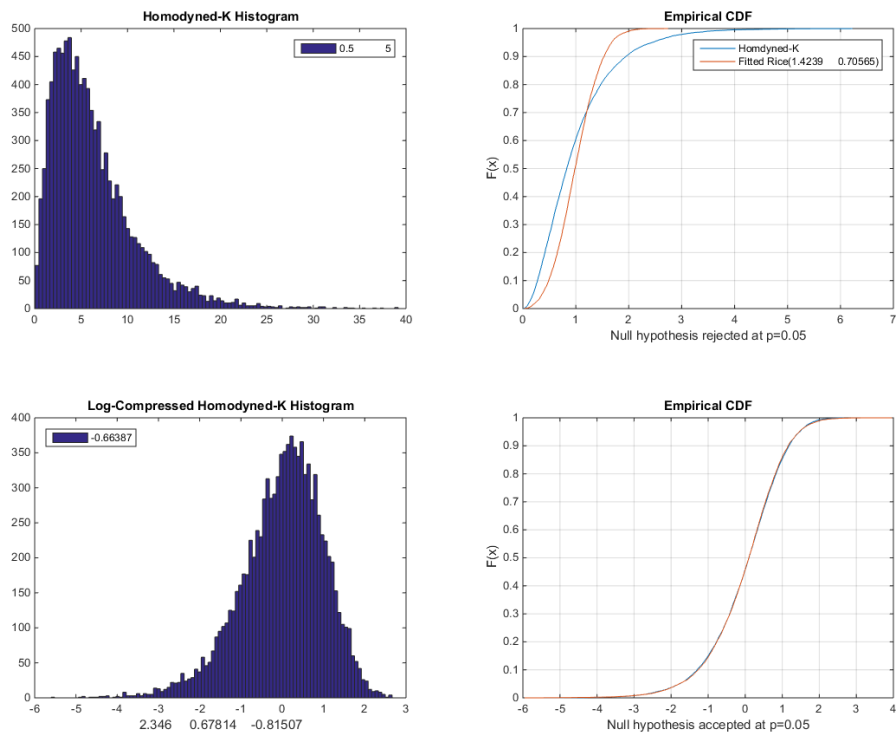


FIGURE 4.8: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Medium effective density $\alpha = 5$ and low coherent component $\nu = 0.5$.

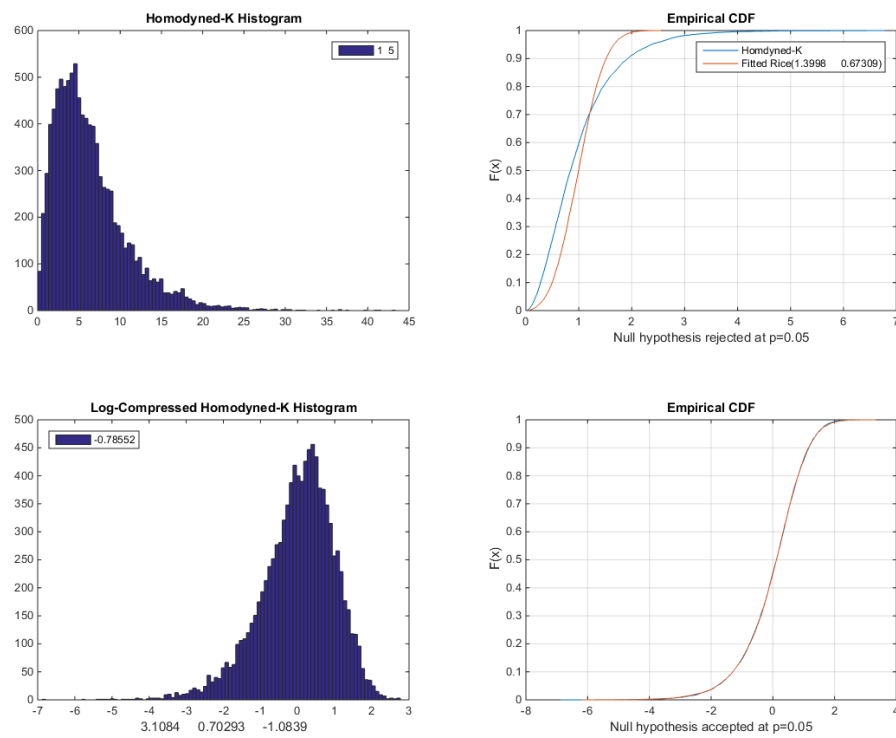


FIGURE 4.9: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Medium effective density $\alpha = 5$ and coherent component $\nu = 1$.

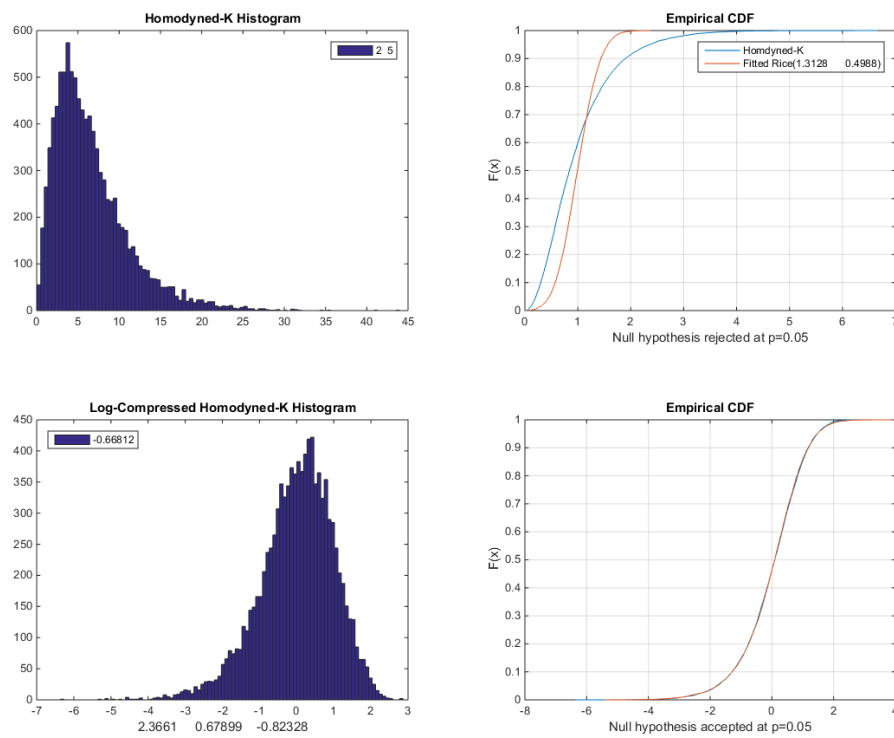


FIGURE 4.10: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Effective density $\alpha = 5$ and coherent component $\nu = 2$.

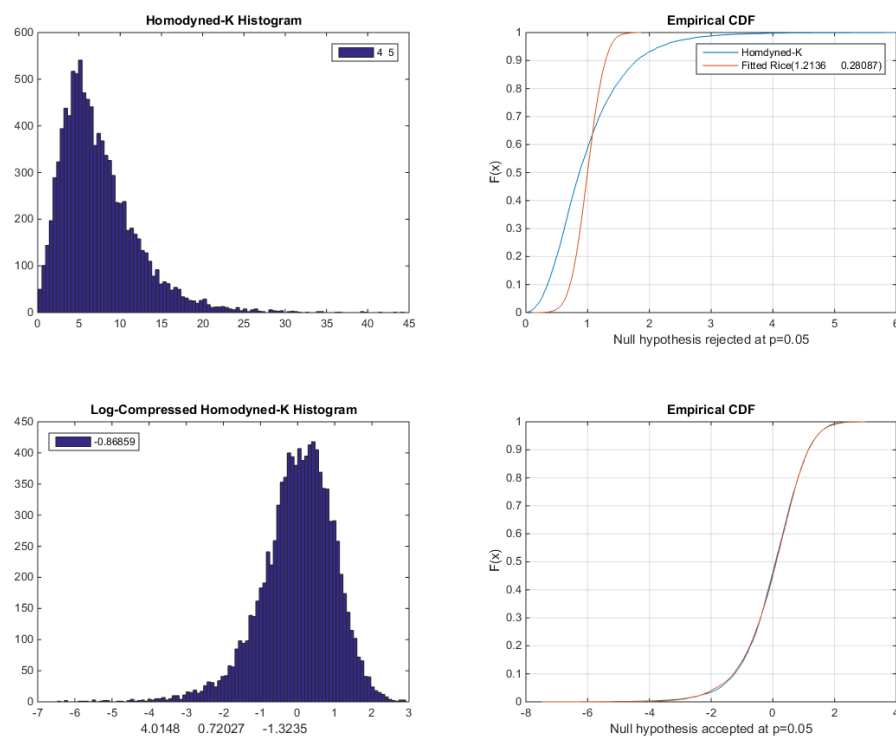


FIGURE 4.11: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Effective density $\alpha = 5$ and coherent component $\nu = 4$.

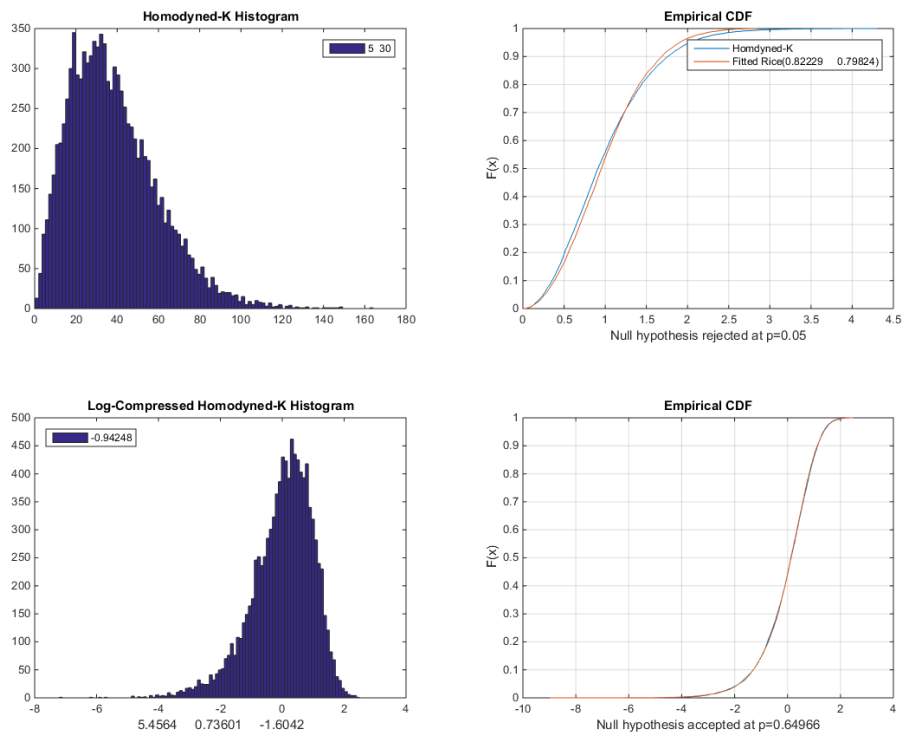


FIGURE 4.12: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Effective density $\alpha = 30$ and coherent component $\nu = 5$

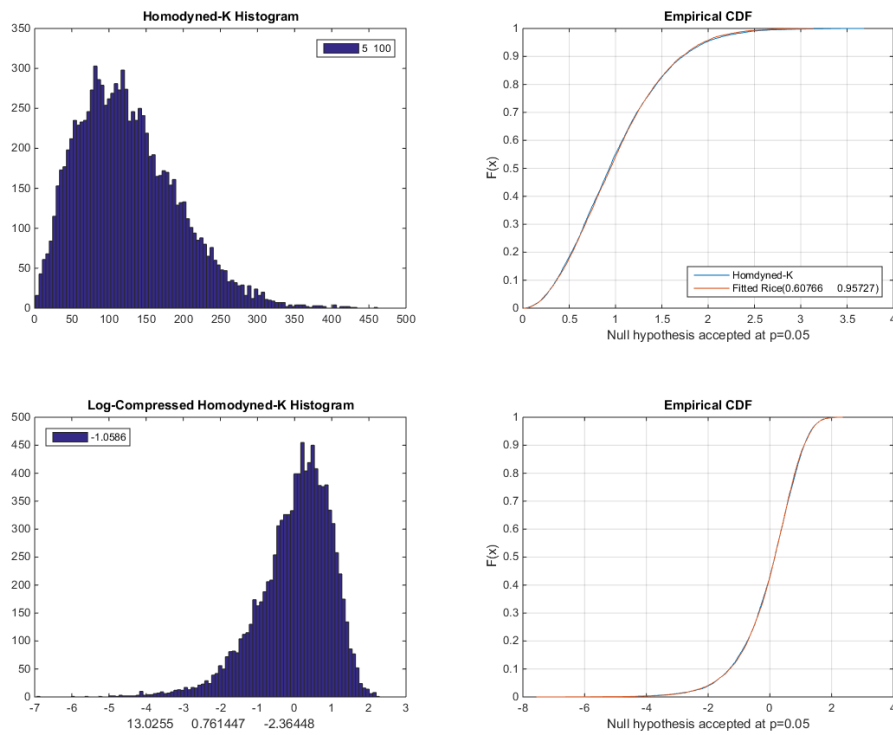


FIGURE 4.13: Montecarlo experiment for testing the HG0 distribution goodness of fit with Homodyned-K distributed data. Effective density $\alpha = 100$ and coherent component $\nu = 5$. Note the goodness of fit with the Rice distribution. Corresponds to the limit $\alpha \rightarrow \infty$ with strong coherent component

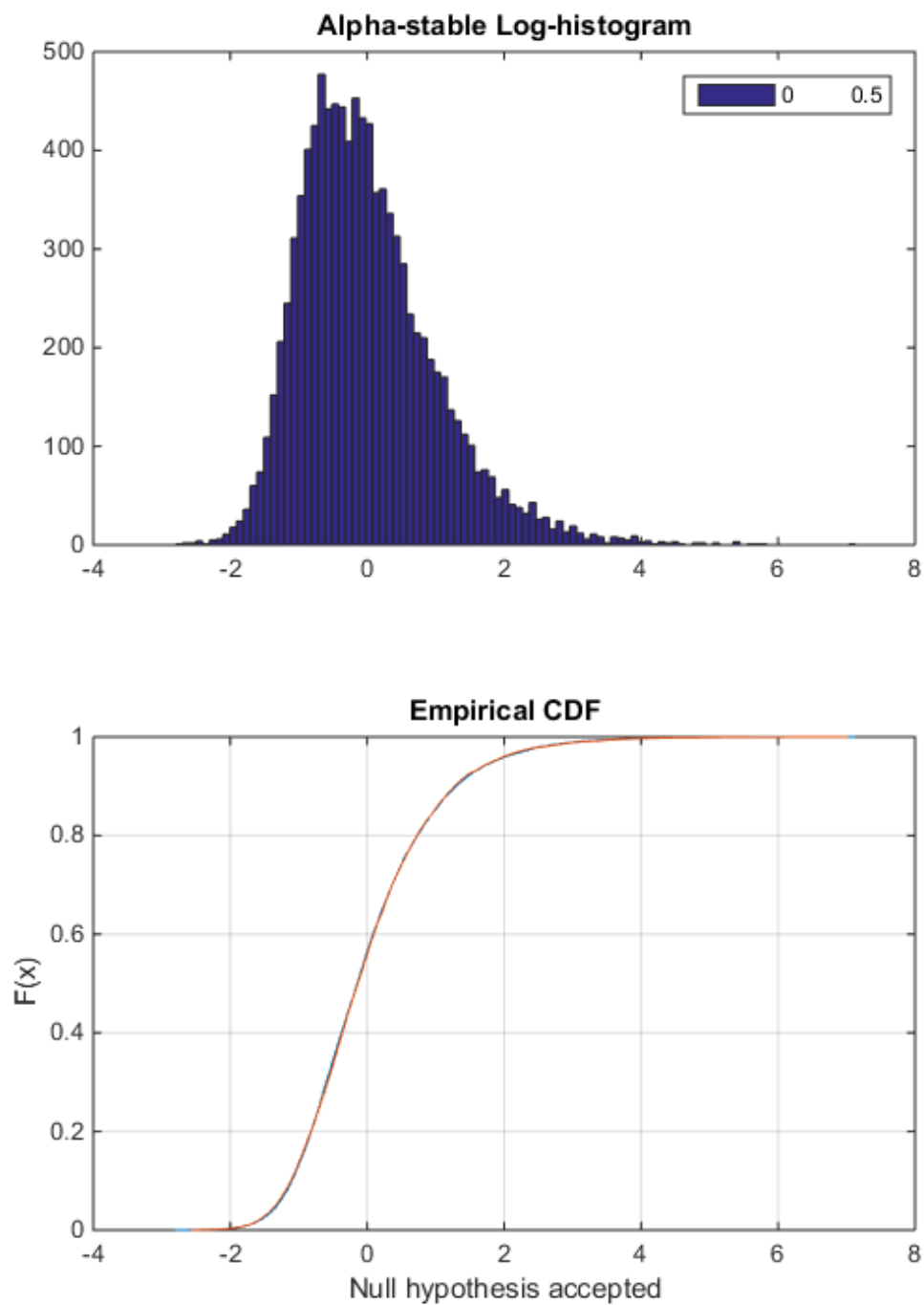


FIGURE 4.14: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 0.5$ and coherent component $\nu = 0$.

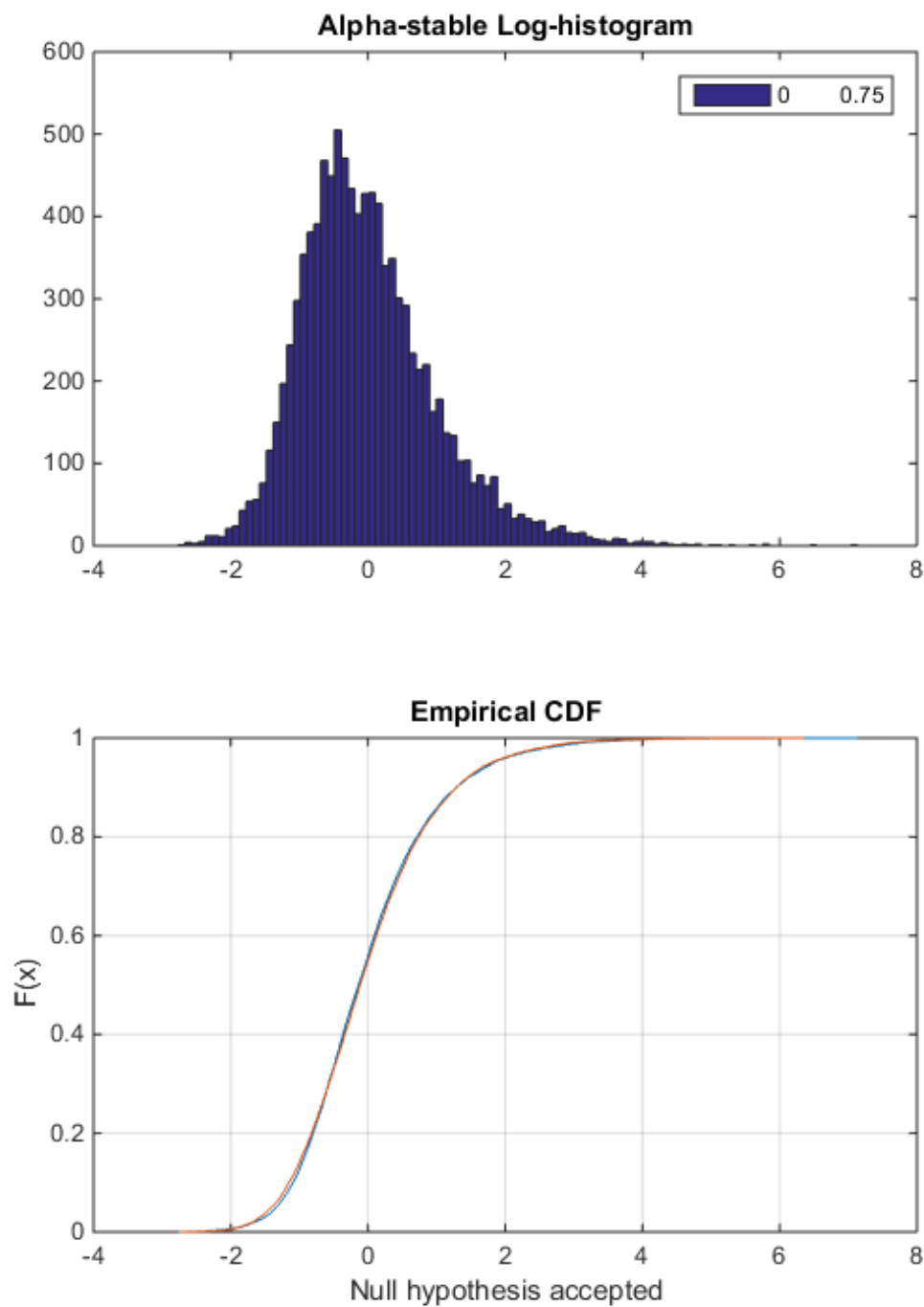


FIGURE 4.15: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 0.75$ and coherent component $\nu = 0$.

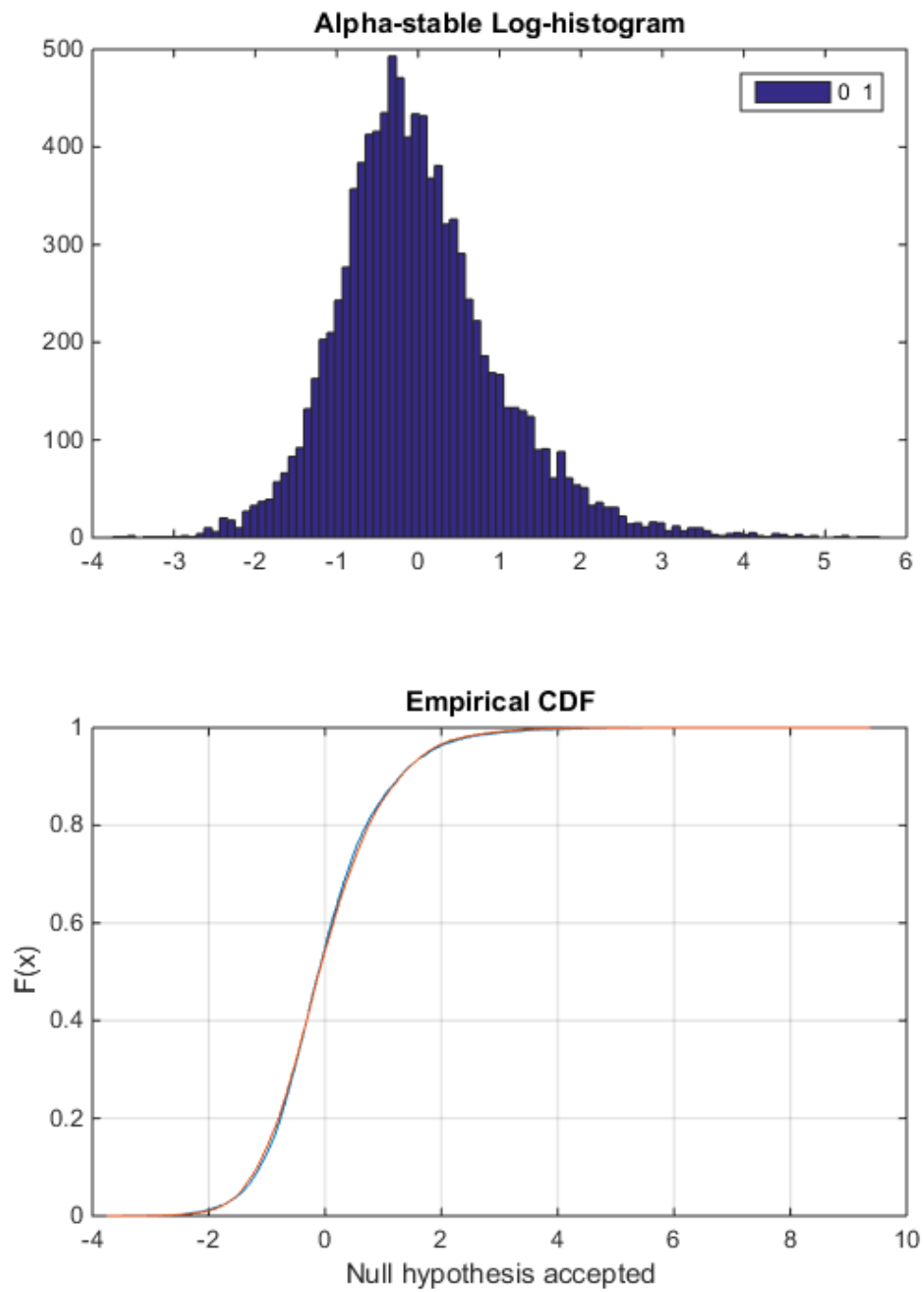


FIGURE 4.16: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 1.0$ and coherent component $\nu = 0$.

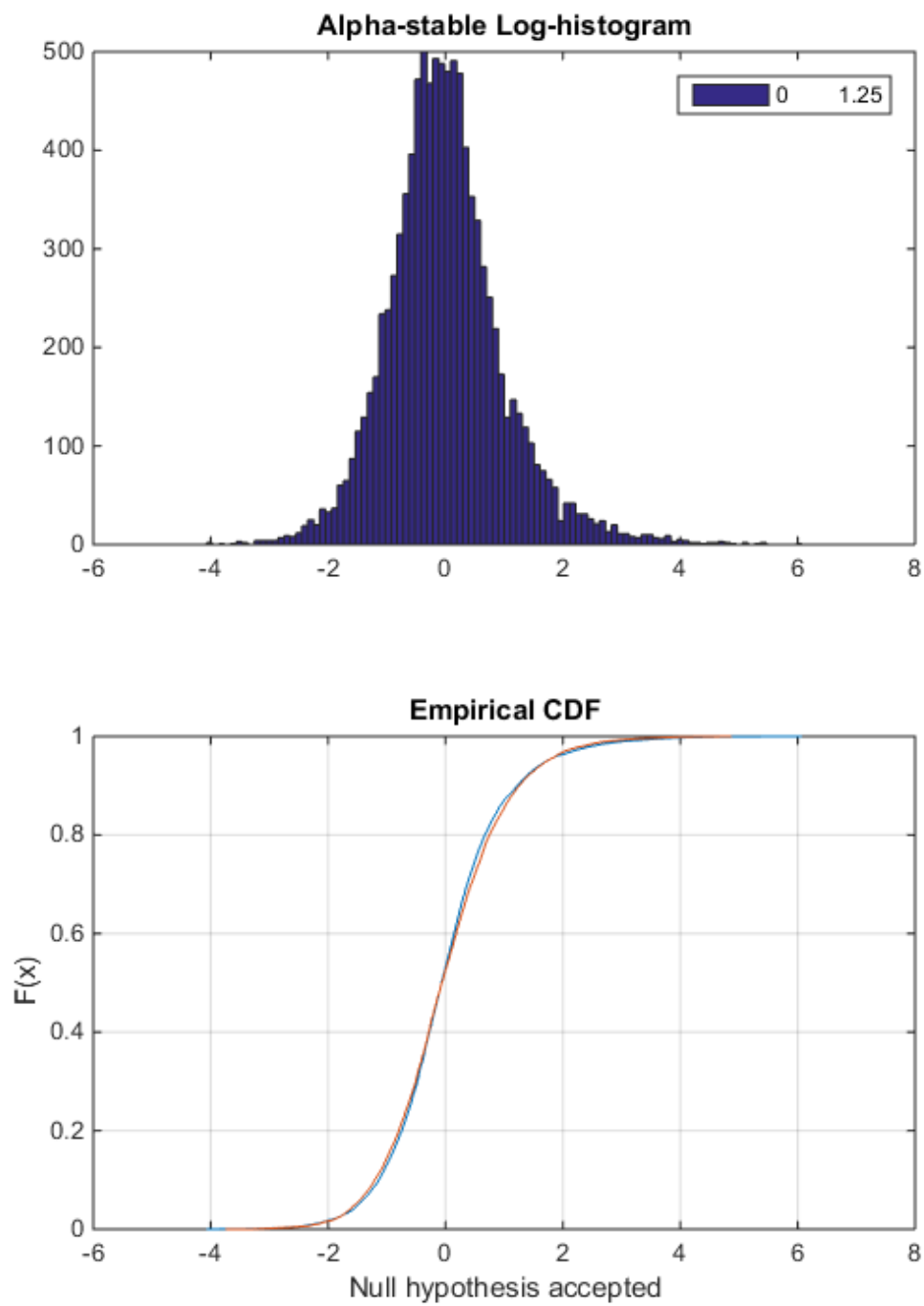


FIGURE 4.17: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 1.25$ and coherent component $\nu = 0$.

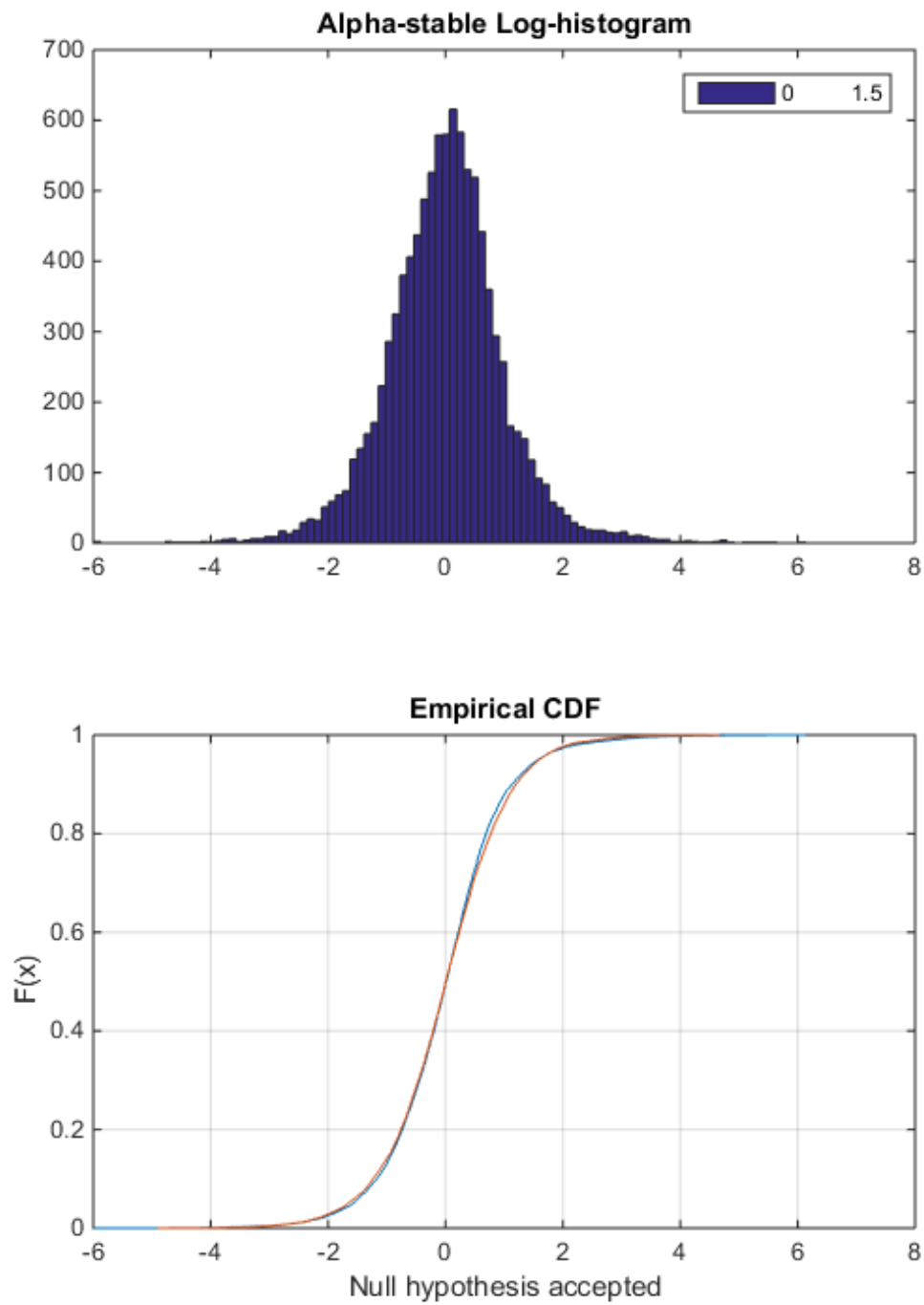


FIGURE 4.18: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 1.50$ and coherent component $\nu = 0$.

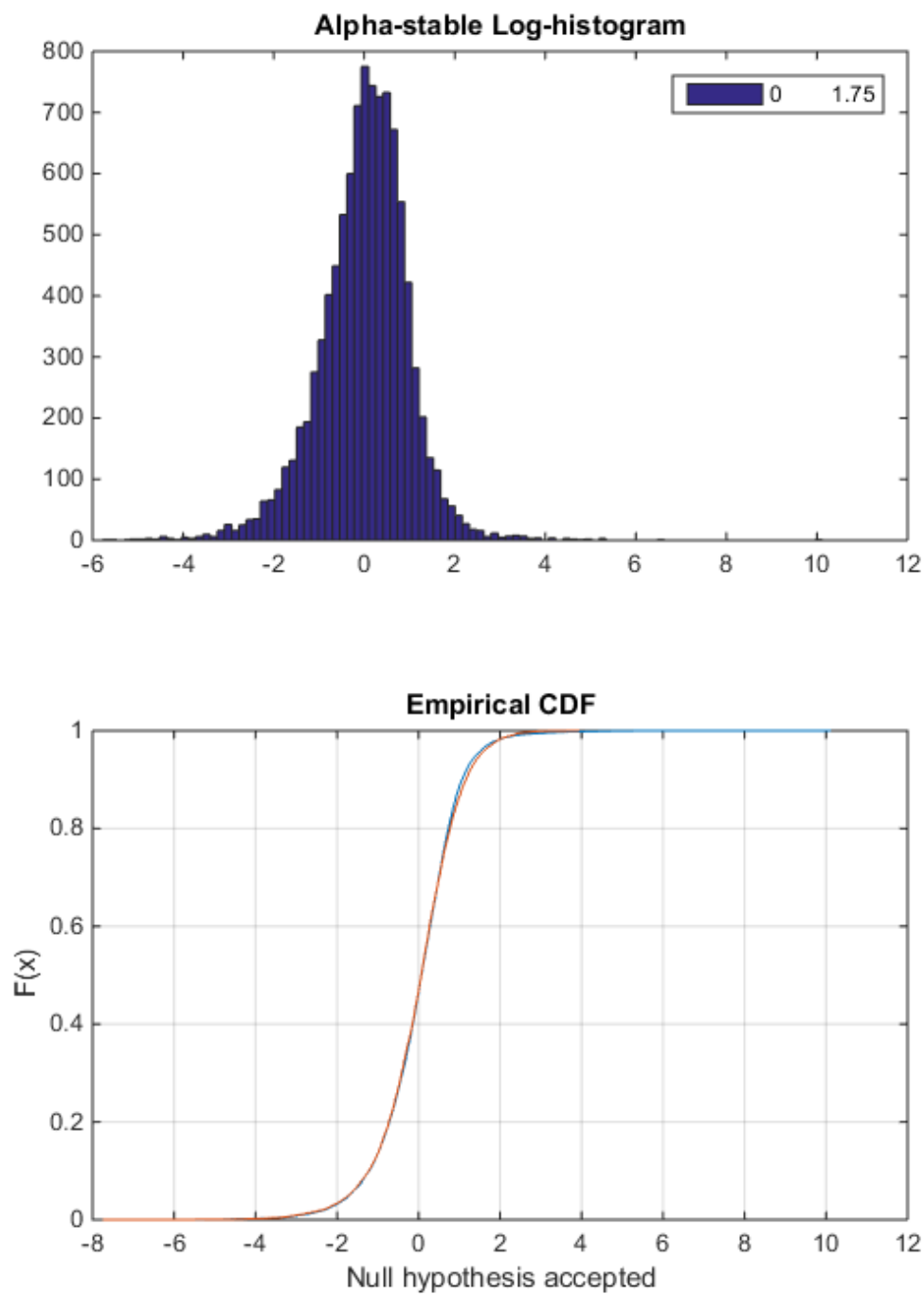


FIGURE 4.19: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 1.75$ and coherent component $\nu = 0$

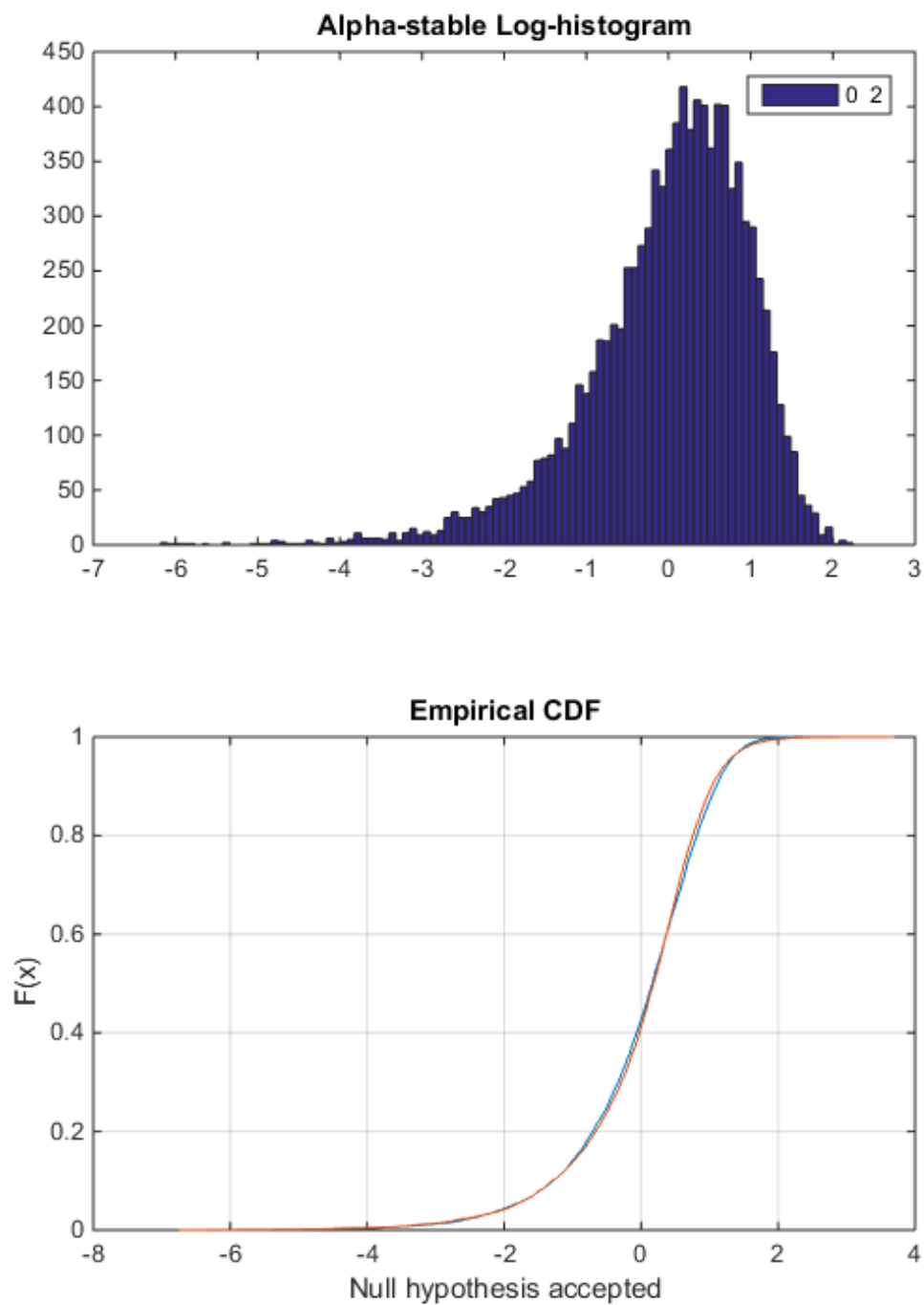


FIGURE 4.20: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 2.0$ and coherent component $\nu = 0$.

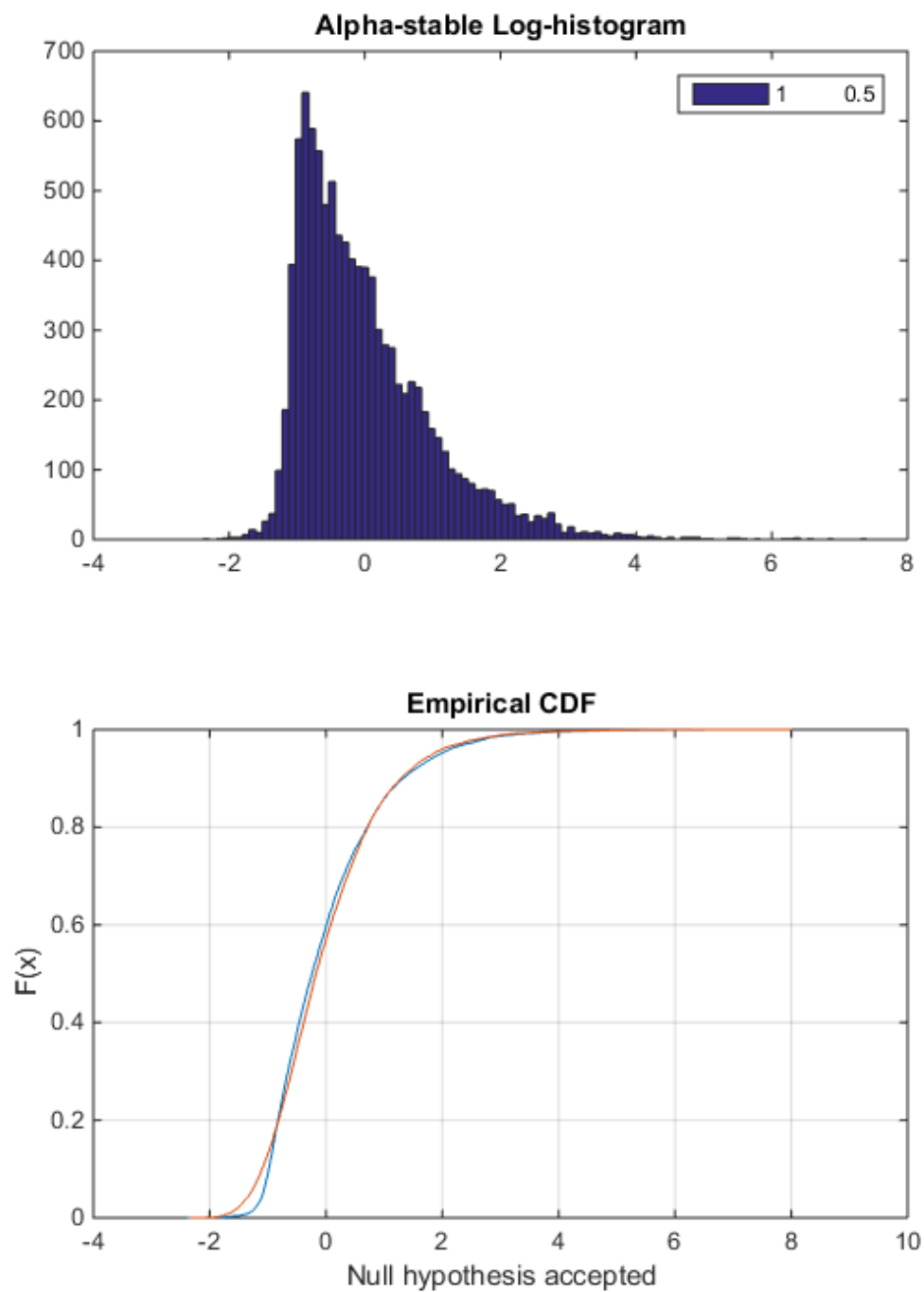


FIGURE 4.21: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 0.5$ and coherent component $\nu = 1.0$

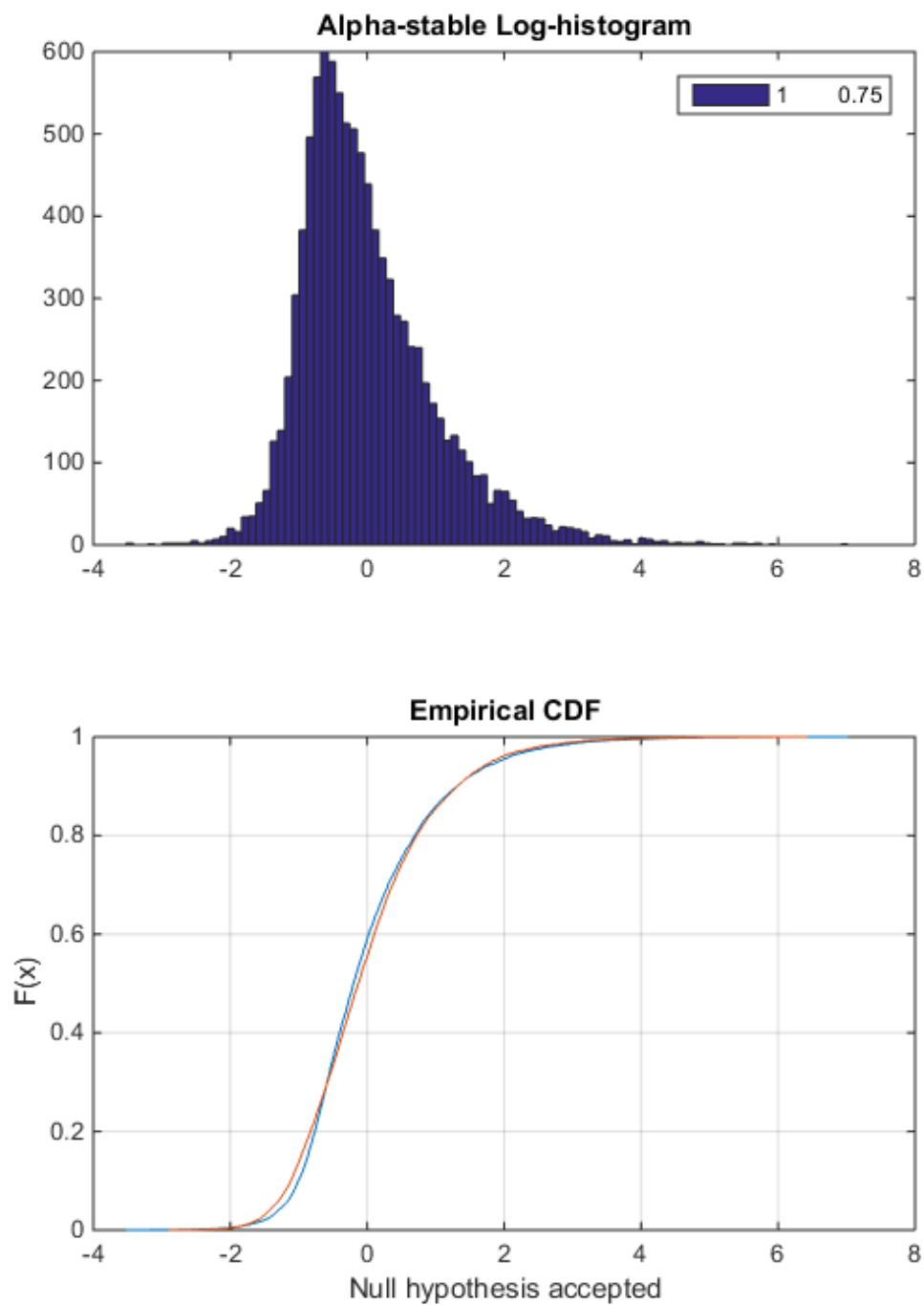


FIGURE 4.22: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 0.75$ and coherent component $\nu = 1.0$

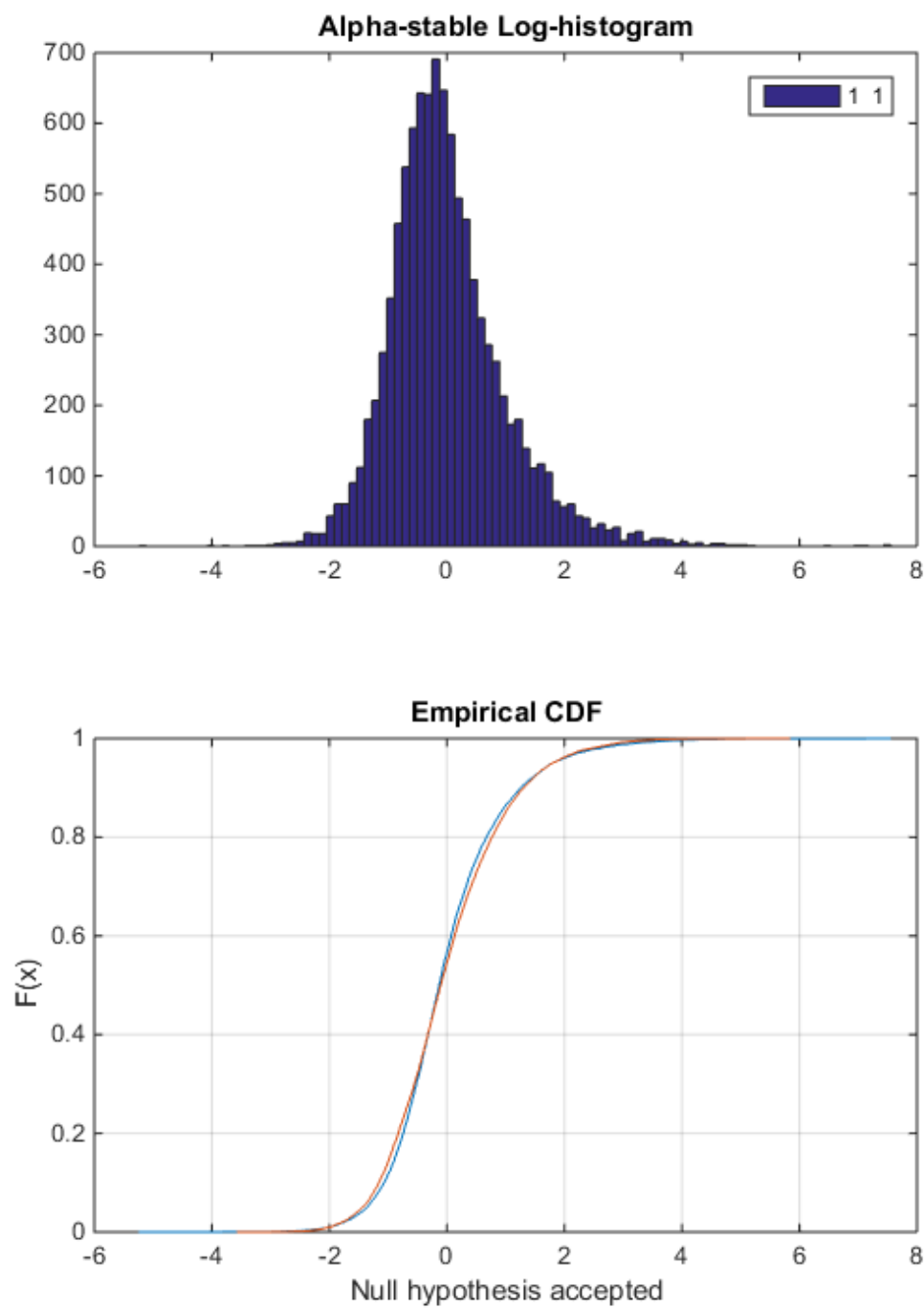


FIGURE 4.23: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 1.0$ and coherent component $\nu = 1.0$

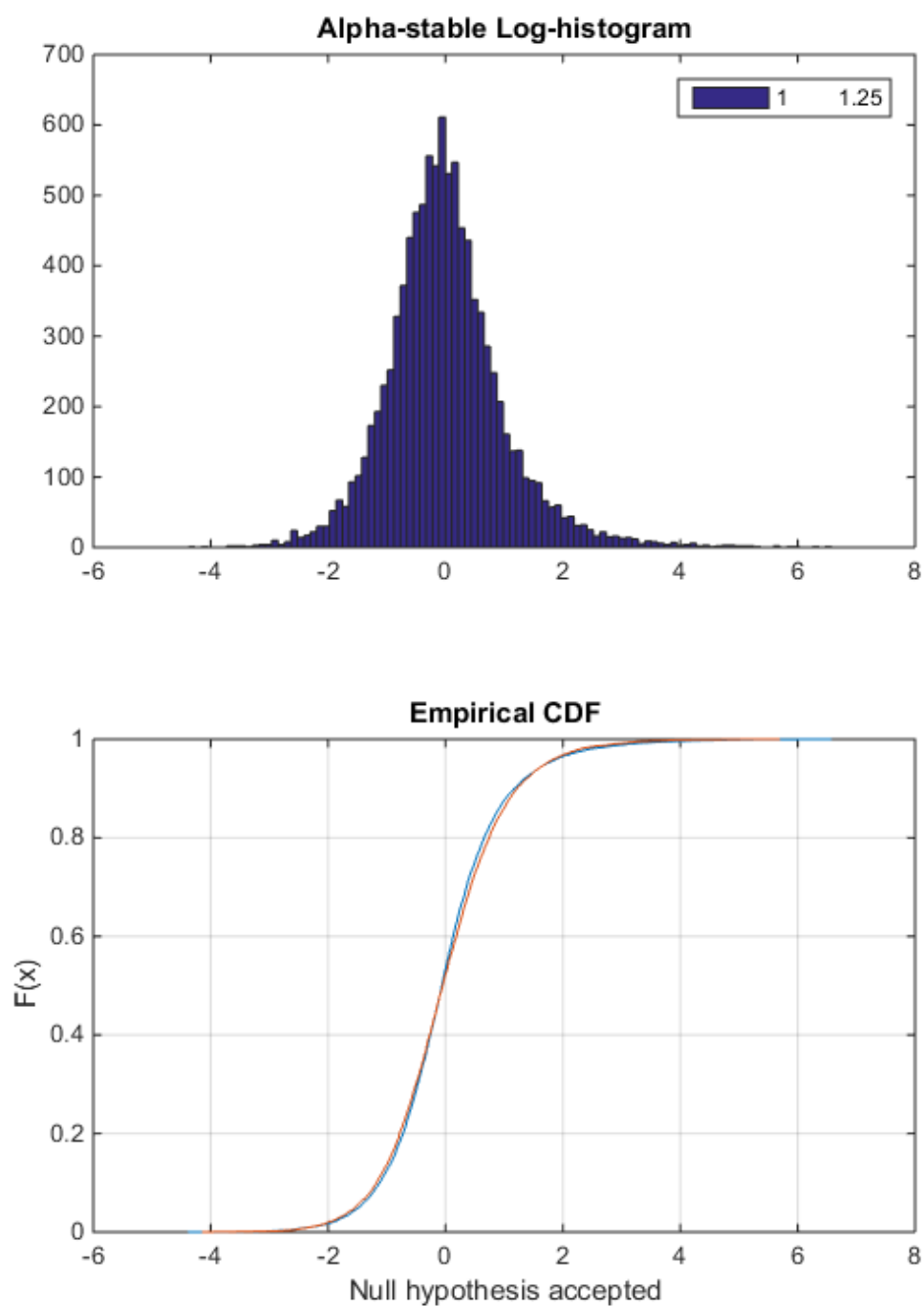


FIGURE 4.24: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 1.25$ and coherent component $\nu = 1.0$

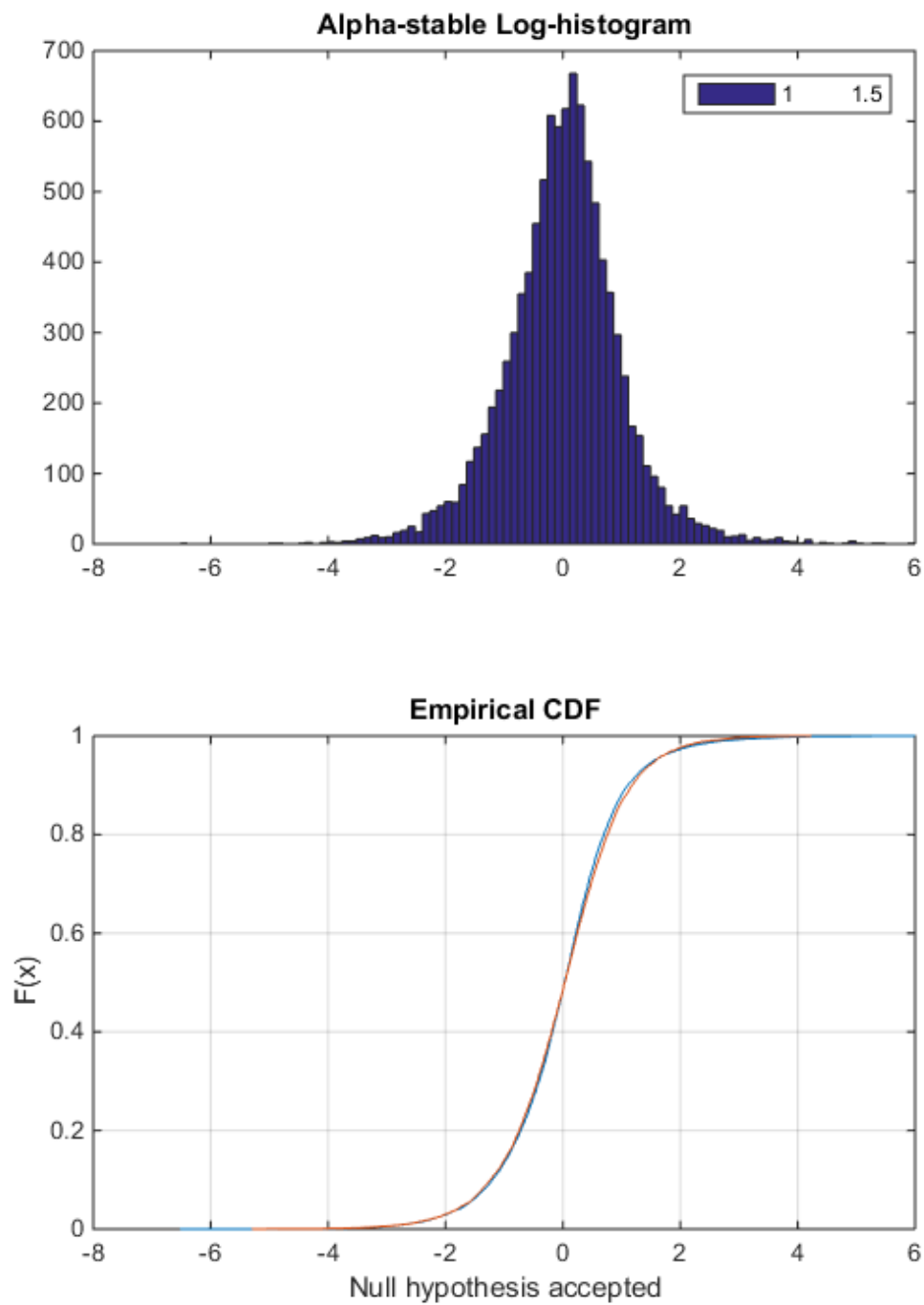


FIGURE 4.25: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 1.5$ and coherent component $\nu = 1.0$

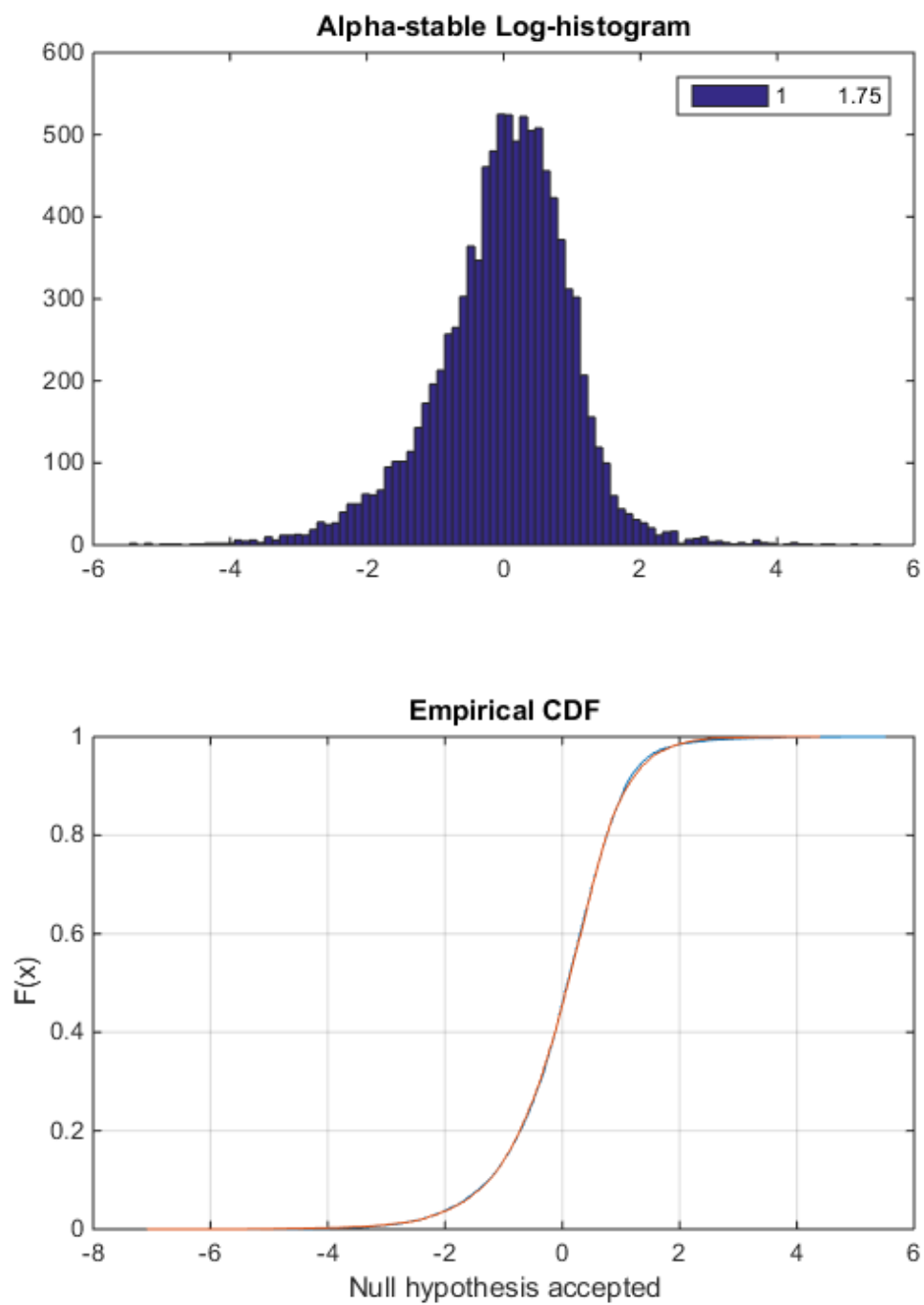


FIGURE 4.26: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 1.75$ and coherent component $\nu = 1.0$

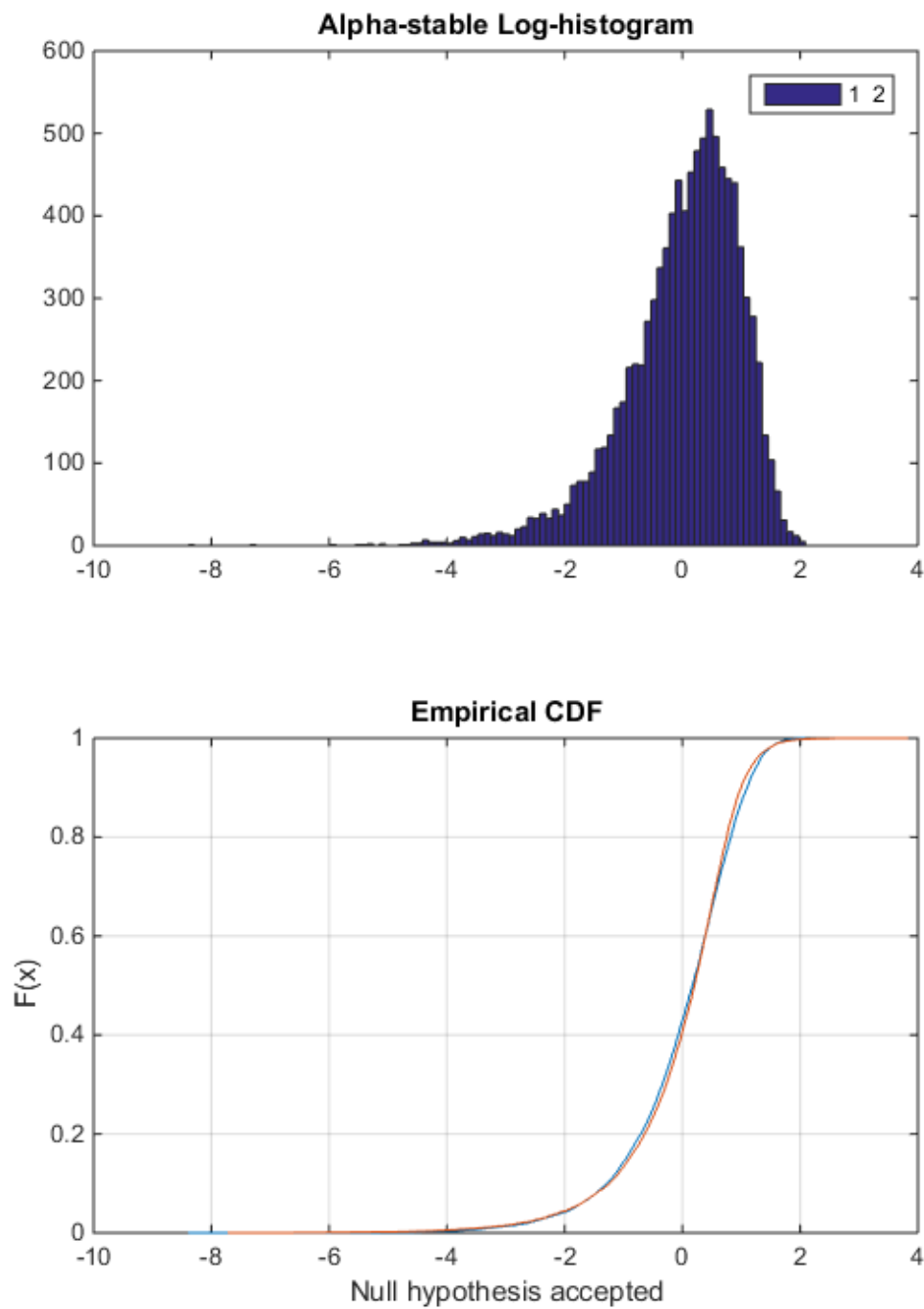


FIGURE 4.27: Montecarlo experiment for testing the HG0 distribution goodness of fit with Log-compressed Alpha-Stable distributed data. Stable exponent $\alpha = 0.5$ and coherent component $\nu = 1.0$

Chapter 5

Central Limit Theorem Revisited

The mathematical theory to demonstrate the generality of the model is presented. It is shown that the developed model has as particular cases all the models presented in chapter 2 when they are Log-compressed.

5.1 Generalized Central limit theorem

The classical Central Limit Theorem says that the normalized sum of independent, identical terms with a finite variance converges to a normal distribution. To be more precise, let X_1, X_2, X_3, \dots be independent identically distributed random variables with mean μ and variance σ^2 . The classical Central Limit Theorem states that the sample mean $X_n = (X_1 + \dots + X_n)/n$ will have

$$a_n(X_1 + \dots + X_n) - b_n \xrightarrow{d} Z \sim N(0, 1), n \rightarrow \infty \quad (5.1)$$

where $a_n = 1/\sigma\sqrt{n}$ and $b_n = \sqrt{n}\mu/\sigma$

The Generalized Central Limit Theorem shows that if the finite variance assumption is dropped, the only possible resulting limits are stable distributions.

A nondegenerate random variable Z is α -stable for some $0 < \alpha \leq 2$ if and only if there is an independent, identically distributed sequence of random variables X_1, X_2, X_3, \dots and constants $a_n > 0, b_n \in \mathbb{R}$ with

$$a_n(X_1 + \dots + X_n) - b_n \xrightarrow{d} Z \quad (5.2)$$

5.2 α -Stable distributions

Stable distributions are a rich class of probability distributions that allow skewness and heavy tails and have many intriguing mathematical properties. The class was characterized by Paul Levy in his study of sums of independent identically distributed terms in the 1920's. The lack of closed formulas for densities and distribution functions for all but a few stable distributions (Gaussian, Cauchy and Levy), has been a major drawback to the use of stable distributions. There are now reliable computer programs to compute stable densities, distribution functions and quantiles. With these programs, it is possible to use stable models in a variety of practical problems.

5.2.1 Definition

Non-degenerate Z is stable if and only if for all $n > 1$, there exist constants $c_n > 0$ and $d_n \in \mathbb{R}$ such that:

$$X_1 + \dots + X_n \stackrel{d}{=} c_n X + d_n \quad (5.3)$$

where X_1, \dots, X_n are independent, identical copies of X . X is strictly stable if and only if $d_n = 0$ for all n .

The most concrete way to describe all possible stable distributions is through the characteristic function, or their Fourier transform. For a strictly stable, its characteristic function is (Nolan, 2015):

$$\phi(u) = \exp(|u|^\alpha) \quad (5.4)$$

5.3 Infinite divisibility and Alpha-stable distributions

A probability distribution is infinitely divisible if it can be expressed as the probability distribution of the sum of an arbitrary number of independent and identically distributed random variables. The characteristic function of any infinitely divisible distribution is then called an infinitely divisible characteristic function. As a result all members of the stable distribution family are infinitely divisible.

The classical example of infinite divisible distribution is the Gamma distribution, and specifically the exponential distribution. The characteristic exponential distribution, can easily be calculated:

$$\phi(t) = \int_0^\infty \exp(-x) \exp(itx) dx = \frac{1}{1 - it}$$

Hence, the sum of k exponential random variables, is a gamma random variable with parameters $(k, 1)$.

5.4 Self-decomposable distributions

A probability distribution is said to be Sum Self Decomposable (henceforth SSD) if its characteristic function satisfies:

$$\phi(u) = \psi(\alpha \cdot u) \cdot \psi_\alpha(u) \quad (5.5)$$

for all $\alpha \in (0, 1)$, with $\psi_\alpha(u)$ a characteristic function. For the corresponding random variable this means that

$$X \stackrel{d}{=} \alpha X' + X_\alpha \quad (5.6)$$

for all $\alpha \in (0, 1)$, where X' and X_α are independent and X' is distributed as X . The class of SSD distributions are the limit laws for the sum of independent but identically distributed (INID) random variables. This property makes SSD random variables attractive for random walk modeling.

5.5 The Kullback–Leibler divergence

The Kullback and Leibler, 1951 divergence is a non-symmetric measure of the difference between two probability distributions P and Q . For distributions P and Q of a positive continuous random variable x , the Kullback–Leibler divergence is defined to be the integral:

$$D_{KL}(P, Q) = \int_0^\infty p(x) \log \frac{p(x)}{q(x)} dx$$

where p and q denote the densities of P and Q .

5.6 G0 and Alpha-stable distributions

A theorem giving the close connection between G0 and Alpha-stable distributions is established and constitutes one of the main contributions of this thesis. A preliminary theorem was originally enunciated by Shanbhag and Sreehari, 1977 and is adapted here to account for the equivalence between G0 and extreme alpha-stable distributions, establishing a connection between Generalized Central Limit Theorem and the Multiplicative Model.

5.7 The main theorem

Theorem 1 *Let Y_α an extreme stable random variable with characteristic exponent α and let Z be an exponential random variable independent of Y_α then,*

$$D \log(Y_\alpha) + G \stackrel{d}{=} HG0(\alpha, d, g), 0 < \alpha < 1$$

5.8 Proof

The proof is divided, for convenience of clarity, in four sections. The first one establishes a connection between the multiplicative model approach to coherent imaging and the G0 distribution in intensity format. Then, the mathematical equivalence between a product of random variables and the G0 distribution in intensity format is given. Next, the multiplicative model representation for strict stable random variable with characteristic exponent α , Y_α , is found. Finally, the log-compressed distribution for Y_α is evaluated.

5.8.1 Multiplicative Model and G0 distribution in Intensity format

The multiplicative model exposed in chapter 3 assumes that the observations within this kind of images are the outcome of the product of two independent random variables: one (X) modeling the terrain backscatter, and other (Y) modeling the speckle noise. The former usually considered real and positive, while the latter could be complex (if the considered image is in complex format) or positive real (intensity and amplitude formats). In order to make an equivalence between the intensity and amplitude formats, the relation between intensity and amplitude i.e. $I = A^2$, must be considered to enunciate the Multiplicative Model in intensity format. As a result, the Amplitude Distribution Function $f_A(A)$ is transformed into $f_I(x) = f_A(\sqrt{x})/(2\sqrt{x})$. Taking into account this relation, the Speckle Amplitude Distribution ($\Gamma^{1/2}(A, 1, 1)$) is transformed into exponential distribution, and the square root of inverse gamma distribution is

transformed into the inverse gamma distribution:

$$2A \exp(-A^2) \rightarrow 2 * \sqrt{x} \exp(-(\sqrt{x})^2)/(2\sqrt{x}) = \exp(-x)$$

$$\frac{2}{\gamma^\alpha \Gamma(\alpha)} A^{2\alpha-1} \exp(-\gamma/A^2) \rightarrow \frac{2}{\gamma^\alpha \Gamma(\alpha)} x^{(2\alpha-1)/2} \exp(-\gamma/(\sqrt{x})^2)/(2\sqrt{x})$$

$$= \frac{1}{\gamma^\alpha \Gamma(\alpha)} x^{\alpha-1} \exp(-\gamma/x)$$

5.8.2 G0 Multiplicative Model representation

As a result, if X and Y are exponential distributed random variables then

$XY^{-1} \sim G0_I(x, 1, 1)$ which is G0 distribution in Intensity format:

$$G0_I(x, 1, 1) = X \cdot Y^{-1} = \frac{1}{(1+x)^2}.$$

5.8.3 G0 Compound representation

Consider the Beta prime distribution of a positive random variable:

$$Bp(x; a, b) = \frac{x^{a-1}}{B(a, b)(1+x)^{a+b}}$$

Where B(a,b) represents the Beta function and a,b>0. It is apparent that the G0 distribution is a particular case of Beta prime distribution with parameters (1, α).

Dubey, 1970 introduced an extension of Beta prime distribution and showed that the following relations are met:

- If $X \sim \Gamma(a, 1)$ and $Y \sim \Gamma(b, 1)$ then $XY^{-1} \sim Bp(a, b)$
- If $X \sim Bp(a, b)$ then $X^{-1} \sim Bp(b, a)$
- $Bp(x; a, b) = \int \Gamma(x; a, p) \Gamma(p; b, 1)$

This relations show that the compound representation and the multiplicative model representation are different aspects of symmetry properties of Gamma distributions.

5.8.4 Shanbhag and Sreehari, 1977 theorem

Let Y_a an extreme stable random variable with characteristic exponent a and let Z be an exponential random variable independent of Y_a then

$$Y_a \stackrel{d}{=} Z \cdot Z^{-1/a}$$

Proof :

$$Y_a \stackrel{d}{=} Z \cdot Z^{-1/a} \implies Z \stackrel{d}{=} Y_a \cdot Z^{1/a}$$

Given $u > 0$ then, by using the exponential distribution:

$$P(Z \geq Y_a u^{1/a}) = \int_0^\infty \exp(-y u^{1/a}) dP(Y_a \leq y)$$

Where $dP(Y_a \leq y)$ is the stable density $f_a(y)dy$:

$$P(Z \geq Y_a u^{1/a}) = \int_0^\infty \exp(-y u^{1/a}) f_a(y) dy$$

Using the stable characteristic function, the stable density is its the inverse Fourier transform:

$$P(Z \geq Y_a u^{1/a}) = \frac{1}{2\pi} \int_0^\infty \exp(-y u^{1/a}) \left[\int_{-\infty}^\infty \exp(-t^a) \exp(-ity) dt \right] dy$$

Changing integration order,

$$P(Z \geq Y_a u^{1/a}) = \frac{1}{2\pi} \int_{-\infty}^\infty \exp(-t^a) \left[\int_0^\infty \exp[-y(u^{1/a} + it)] dy \right] dt$$

$$P(Z \geq Y_a u^{1/a}) = \frac{1}{2\pi} \int_{-\infty}^\infty \frac{\exp(-t^a)}{u^{1/a} + it} \left[\exp[-y(u^{1/a} + it)] \Big|_{y \rightarrow \infty}^0 \right] dt$$

$$P(Z \geq Y_a u^{1/a}) = \frac{1}{2\pi} \int_{-\infty}^\infty \frac{\exp(-t^a)}{u^{1/a} + it} dt$$

Using the Residue Theorem,

$$P(Z \geq Y_a u^{1/a}) = \frac{2\pi i}{2\pi} \text{Res} \left(\frac{\exp(-t^a)}{u^{1/a} + it} \right)$$

$$P(Z \geq Y_a u^{1/a}) = \frac{2\pi i}{2\pi i} \exp(-u^{1/a})^a$$

$$P(Z \geq Y_a u^{1/a}) = \exp(-u)$$

$$Y_a \cdot Z^{1/a} \stackrel{d}{=} Z \implies Y_a \stackrel{d}{=} Z \cdot Z^{-1/a}$$

5.8.5 Stable distributions in Amplitude format

Taking into account the relation between Intensity and Amplitude i.e. $I = A^2$, then

$$\begin{aligned} Y_a^{1/2} &\stackrel{d}{=} Z^{1/2} \cdot Z^{-1/2a} = \Gamma^{1/2} \cdot \Gamma^{-1/2}(a, 1) \\ Y_a^{1/2} &\stackrel{d}{=} G0(a, 1) = G0(n\alpha, 1) \quad n = 2, \quad 0 < \alpha \leq 1 \end{aligned}$$

5.8.6 Log-compression of stable distributions

Consider a Log compression of a stable random variable:

$$\Psi = D \cdot \log(Y_a(x)) + G$$

From the previous result it follows:

$$\begin{aligned} \Psi &= D \cdot \log(Z^{1/2} \cdot Z^{-1/2a}) + G \\ \Psi &\sim HG0(\alpha, d, g) \quad 0 < \alpha \leq 1 \end{aligned}$$

5.9 The multiplicative model and the Generalized Central Limit Theorem

(Shanbhag and Sreehari, 1977) probed that if Z_r is random variable distributed according to a gamma distribution with index parameter $0 < r \leq 1$ and W is a random variable independent of Z , then for every $p \geq 1$ the random variable $Z_r^p \cdot W$ has an infinitely divisible distribution and can be represented with Alpha-stable distribution or, equivalently with a G0 distribution.

A direct consequence of last result, is that every multiplicative model with noise represented as a Rayleigh distribution is a infinitely divisible distribution and can be described with G0 distribution.

5.10 Compound representation and the multiplicative model

The previous section stated the connection between the Generalized Central limit theorem and the multiplicative model with speckle noise modeled as $\Gamma^{1/2}$ distribution. Now the objective is to account for the connection between the compound representation and the multiplicative model. Let a random variable X with a compound representation that can be expressed as:

$$X \sim \int F(x/w)G(w)dw$$

This means, the random variable X is F distributed with a scale parameter w which is a random variable G distributed. Now, using the relations:

$$z = \frac{x}{w} \quad w = \frac{x}{z} \quad dw = \frac{x}{z^2} dz$$

It is obtained:

$$X \sim \int \frac{1}{w} F(z)G(w)dw$$

$$X \sim \int \frac{z}{x} F(z)G\left(\frac{x}{z}\right) \frac{x}{z^2} dz$$

Therefore,

$$X \sim \int \frac{1}{z} F(z) G\left(\frac{x}{z}\right) dz$$

As a result, if $Z \sim F$ and $W \sim G$ then $X = Z \cdot W$

5.11 Concluding remarks

In this chapter it has been shown that a stable random distribution converges in distribution to G0 distribution and more precisely, the general central limit theorem, the compound representation and the multiplicative model are mathematically equivalent to define stable random variables as limiting distribution of random walks in complex plane. As a result, most of the proposed models up to date, are particular cases of the HG0 distribution when a Log-compression transform is applied to RF envelope amplitude signal.

Chapter 6

Applications

Some practical applications of the new statistical model are presented in this chapter.

6.1 B-scan image filtering.

The un-sharp masking filter smooths the image based on some local statistic. The output Y of an un-sharp masking filter for input X is given by

$$Y = \bar{X} + c(X - \bar{X}) \quad (6.1)$$

where c is the local statistic and \bar{X} is the local mean. If the statistic is limited to range $[0, 1]$, then the filter output will range from maximal smoothing (mean) to no filtering. Using the parameter, α , one can design an un-sharp masking filter similar to the filter proposed by Berger, 1991

Figure 6.1 shows the results of this implementation.

6.2 B-scan image segmentation

In echo-cardiographic images, blood has a very low effective scatterers density, while muscle presents an evident specular reflection. Also, interfaces present a typical fully developed speckle that can be efficiently segmented in figure 6.2

6.3 Heart ejection fraction estimation

The ejection fraction (EF) is an important measurement in determining how well heart is pumping out blood and in diagnosing and tracking heart failure. It is a measurement of how much blood the left ventricle pumps out with each contraction.

An ejection fraction of 60 percent means that 60 percent of the total amount of blood in the left ventricle is pushed out with each heartbeat. It is defined in terms of Diastolic and Systolic volumes as:

$$EF = [(V_{dia} - V_{sis})/V_{dia}] \cdot 100 \quad (6.2)$$

A normal heart's ejection fraction may be between 55 and 70.

A measurement under 40 may be evidence of heart failure or cardiomyopathy.

An EF between 40 and 55 indicates damage, perhaps from a previous heart attack, but it may not indicate heart failure. In severe cases, EF can be very low.

EF higher than 75 percent may indicate a heart condition like hypertrophic cardiomyopathy.

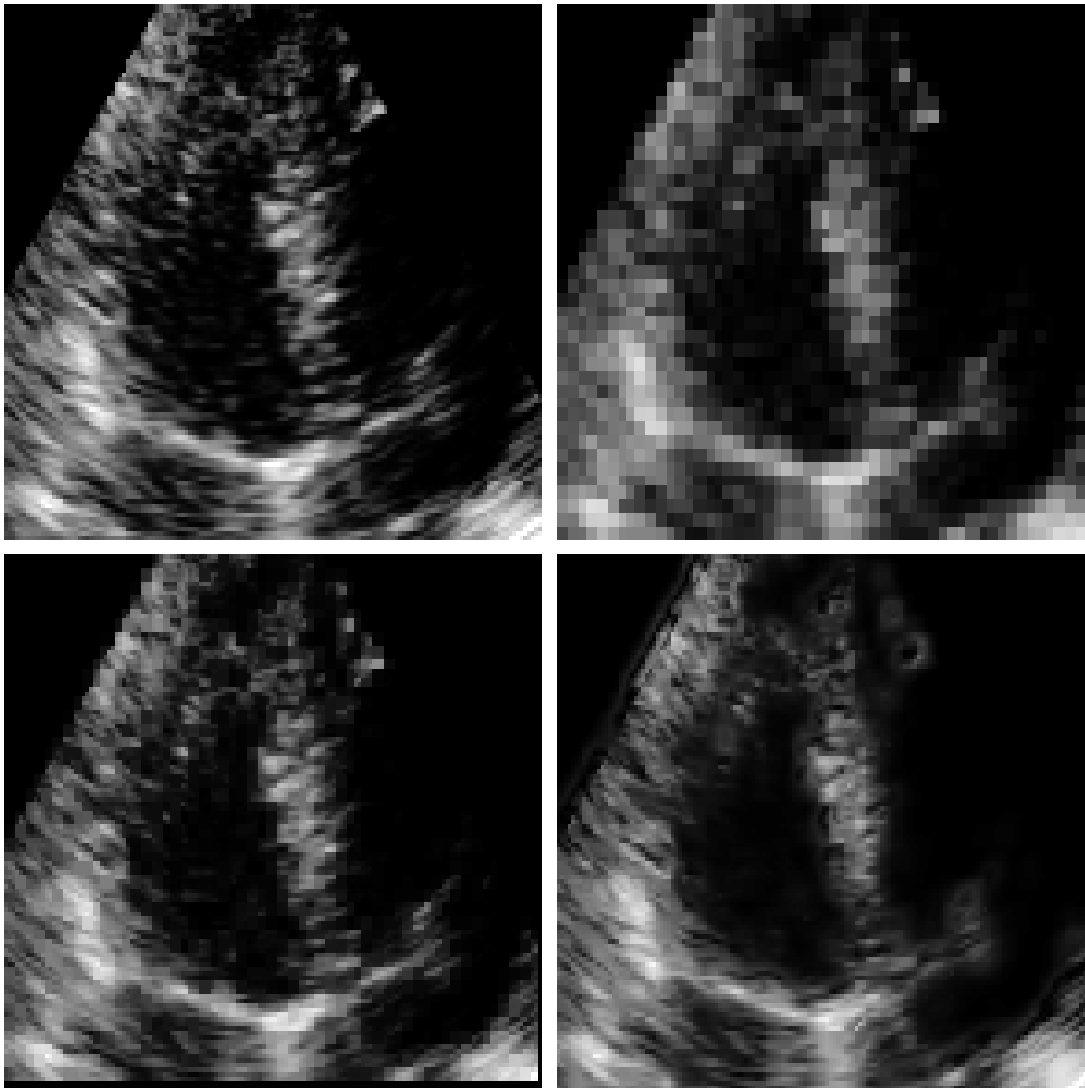


FIGURE 6.1: Ultrasound B-scan image filtering. Upper left: original image. Upper right: mean filter. Bottom left: fix window filter. Bottom right: moving window filter

One direct application of the developed statistical model is just an estimation of the proportion of blood region and muscle regions during a cardiac cycle. Figure 6.3 shows an estimation of the ejection fraction compared with the assisted EF measurement.

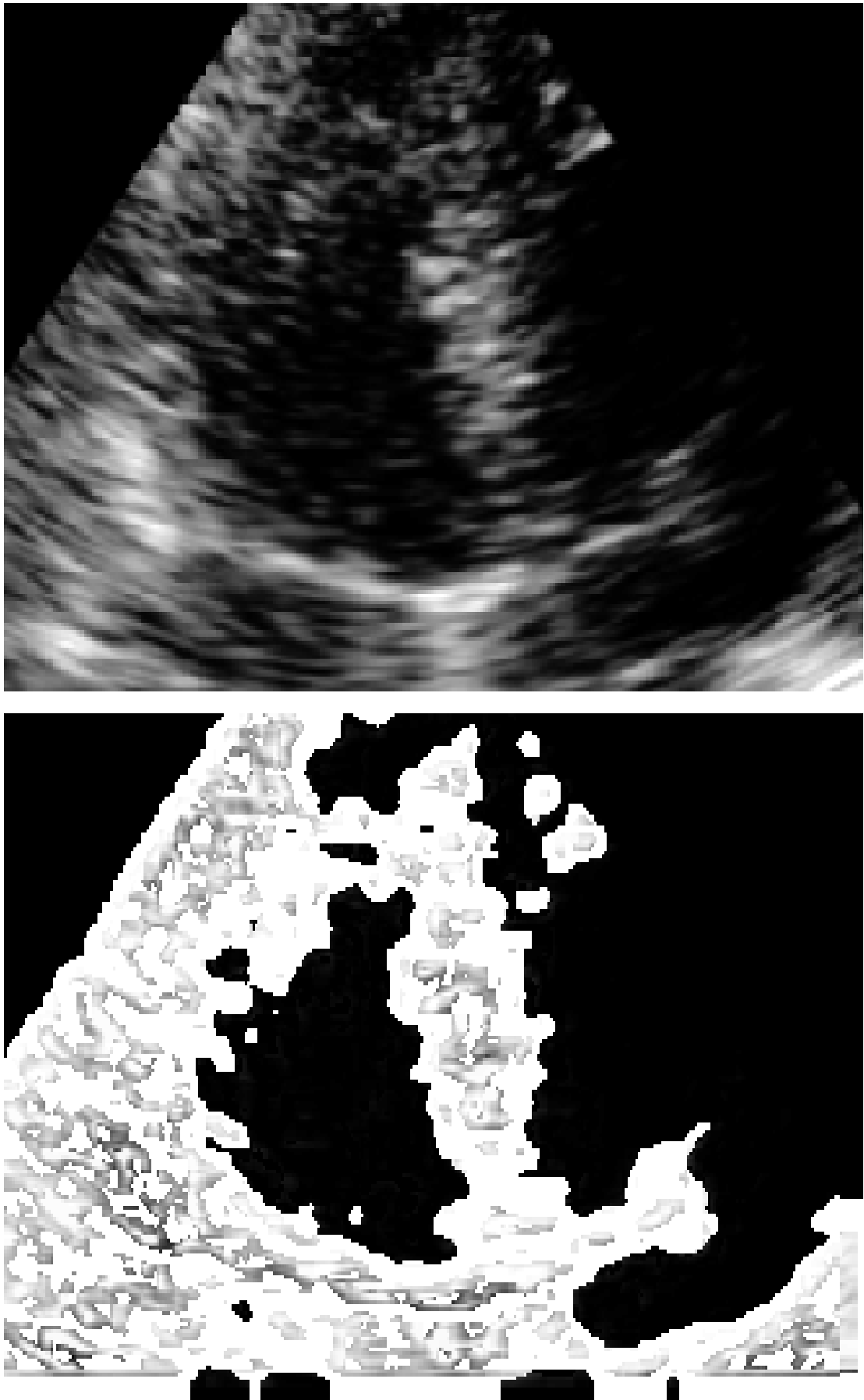


FIGURE 6.2: Ultrasound B-scan image segmentation

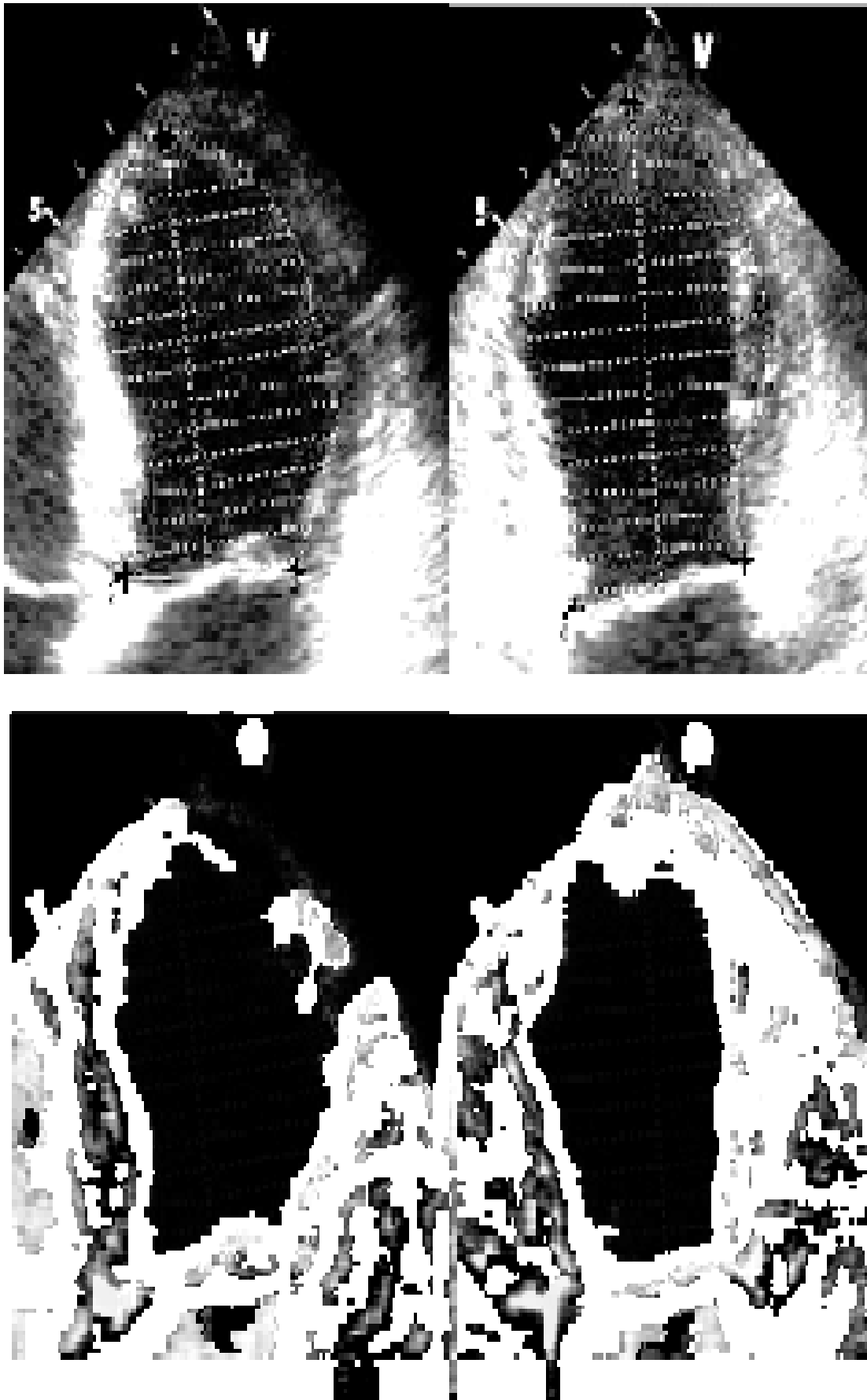


FIGURE 6.3: Ejection Fraction estimation
Estimated 52 ± 10
Assisted 50 ± 4

Chapter 7

Conclusions and future work

7.1 Conclusions

- A new statistical model for Ultrasound B-scan images is developed.
- It is shown that the new model includes as a particular cases, after Log-compression, most of the models proposed up-to-date .
- The new model parameters include the Log-amplifier parameters, and can be estimated with standard parameter estimation methods.
- The new model is applied successfully to:
 - Speckle noise identification and filtering.
 - Ultrasound B-scan image segmentation.
 - Heart ejection fraction (EF) evaluation

7.2 Future Work

- Improving parameter estimation is a initial work towards on-line implementation of filtering, segmentation and EF measurement.
- A clinical study should validate these applications by using standard precision measurement i.e. Magnetic Resonance.
- Real-time measurements, once validated the technology, could improve costs of clinical examinations because of the lower cost of ultrasound equipment.
- A wide Theoretical research can be done extending and applying the developed model to other scientific areas: physics, financial, actuary, statistics, etc.

Appendix A

Moments generating function for HG0 distribution

$$m_X(t) = \langle \exp(tX) \rangle = \int_S \exp(tX) P_X(x) dx$$

$$m_X(t) = \langle \exp(tz) \rangle = \frac{\alpha}{d} \int_{-\infty}^{\infty} \frac{\exp(tz) \exp\left(\frac{z-g}{d}\right)}{\left[1 + \exp\left(\frac{z-g}{d}\right)\right]^{\alpha+1}}$$

Using a variable change:

$$u = 1 + \exp\left(\frac{z-g}{d}\right)$$

$$du = (1/d) \exp\left(\frac{z-g}{d}\right)$$

$$\exp(z) = (u-1)^{td} \exp(gt)$$

$$m_X(t) = \alpha \exp(gt) \int_1^{\infty} \frac{(u-1)^{td}}{u^{\alpha+1}} du$$

Now, with the new variable change $v = 1/u$

$$m_X(t) = \alpha \exp(gt) \int_0^1 (1-v)^{1+td-1} v^{\alpha-td-1} dv$$

Using the Beta Function definition it is got finally

$$m_X(t) = \frac{\exp(gt)}{\Gamma(\alpha)} \Gamma(1+td) \Gamma(\alpha-td)$$

Appendix B

MATLAB code used in this work

B.1 Simulating and plotting K-distributed data

B.1.1 PDF and CDF evaluation

```
function [y,yc]=Kcdf(a,x)
la=(gamma(1.5)*gamma(a+0.5)/gamma(a))^2;
y=((4*la*x)/gamma(a)).*((la*x.^2).^(a-1)/2)).*...
besselk(a-1,(2*sqrt(la)*x));
yc=cumsum(y/sum(y));
```

B.1.2 Plotting PDF and CDF

```
Va=[0.6,1,2,4,8,20];
Vta=num2str(Va);
Vc=['b','r','g','c','m','k'];
x=linspace(0.02,10,500);
figure(2);clf;
for i=1:6
    a=Va(i);c=Vc(i);
    [y1,y2]=Kcdf(a,x);
    xp=x(1:250);yp=y1(1:250);
    figure(2);subplot(2,1,1);hold on;

    plot(xp,yp,c);hold off;grid on;
    figure(2);subplot(2,1,2);hold on;
    subplot(2,1,2);hold on;grid on;

    xp=x(1:250);yp=y2(1:250);
    plot(xp,yp,c);hold off;
end
subplot(2,1,2);

ga=gamma(1.5)^-2;
yr=(2*x/ga).*exp((-x.^2)/ga);xp=x(1:250);yp=yr(1:250);
subplot(211);hold on;plot(xp,yp,'--k');grid on;hold off;
legend('\alpha = 0.6','\alpha =1.0','\alpha =2.0',...
'\alpha =4.0','\alpha =8.0','\alpha =20','Rayl');
yr=cumsum(yr/sum(yr));xp=x(1:250);yp=yr(1:250);
subplot(212);hold on;plot(xp,yp,'--k');grid on;hold off;
legend('\alpha = 0.6','\alpha =1.0','\alpha =2.0',...
```

```
'\alpha =4.0','\alpha =8.0','\alpha =20','Rayl');
```

B.2 Simulating and plotting Homodyned K-distributed data

B.2.1 Simulating Homodyned-K random numbers

```
function ImK= Gkcrnd(N,ro,a)

% Y=ones(N,N);
Sg=random('gam',a,1,100,100);
Ro=ro*ones(N,N);
Y=random('rician',Ro,Sg);
ImK=Y(:);
```

B.2.2 Montecarlo experiments

```
function monteC(b,a)

pk=[b,a];
%Px=Gkrnd(100,b,a) + eps;
Px=Gkcrnd(100,b,a);
%[q]=prctile(Px,[10,90]);mn=q(1);mx=q(2);
Pb=Px/mean(Px);
s2=mean(Pb.^2);s4=mean(Pb.^4);
ep=sqrt(sqrt(abs(2*s2^2-s4)));
sg=sqrt(abs(s2-ep^2));

%rp=[0,sg];Pr=raylrnd(1,1,10000);
rp=[ep,sg];Pr=random('rician',ep,sg,1,10000);
mu=mean(Pr);Pr=Pr/mu;
mu=mean(Px);Pt=Px/mu;
Ho=kstest2(Pr,Pt,'Alpha',0.01);

if Ho==0
lbl='Null hypothesis accepted at p=0.05';
else
lbl='Null hypothesis rejected at p=0.05';
end
mn=min(Px);mx=max(Px);

D=255/log(mx/mn); G = -D*log(mn);
Lx=(D*log(Px) + G);
Lx=double(Lx);
mu=mean(Lx);

z=Lx-mu; sg=std(z);z=z/sg;
[Pt,par]=Hg0p1(z);
Nb=100;
figure(2);clf;
```

```

subplot(221);hist(Px,Nb);
title('Homodyned-K Histogram');
legend(num2str(pk),'Location','NE');
%subplot(222);hist(Lx,Nb);
subplot(222);cdfplot(Px/mean(Px));
hold on;cdfplot(Pr/mean(Pr));hold off;
legend('Homodyned-K',strcat('Fitted Rice(',num2str(rp),')'),...
'Location','SE');
xlabel(lb1);
subplot(223);hist(z,Nb);
xlabel(num2str(par));
title('Log-Compressed Homodyned-K Histogram')

m3=mean(z.^3);
legend(num2str(m3),'Location','NW');

[Ho,p]=kstest2(z,Pt,'Alpha',0.1);

if Ho==0
    lbl=strcat('Null hypothesis accepted at p=',num2str(0.05));
else
    lbl=strcat('Null hypothesis rejected at p=',num2str(0.05));
end

%title(num2str(m3));

subplot(224);cdfplot(z);
hold on;cdfplot(Pt);hold off;
xlabel(lb1);

```

B.2.3 Polynomial fit for HG0 parameters estimation

```

Va=linspace(1,20,191);
m3=(psi(2,1)-psi(2,Va))./((psi(1,1)+psi(1,Va)).^1.5);
p1=polyfit(1./Va,m3,3);
y1=polyval(p1,1./Va);
D=1./sqrt(psi(1,1)+psi(1,Va));
G=D.*(psi(1)-psi(Va));
%plot(1./Va,m3)
im3=(psi(2,1)-psi(2,1./Va))./((psi(1,1)+...
psi(1,1./Va)).^1.5);
p2=polyfit(1./Va,im3,5);
y2=polyval(p2,1./Va);

%plot(1./Va,im3)
Di=1./sqrt(psi(1,1)+psi(1,1./Va));
Gi=Di.*(psi(1)-psi(1./Va));

```

```

z=linspace(-10,10,201);

figure(1);clf;
subplot(221);plot(1./Va,m3,'-');
hold on;plot(1./Va,y1,'r');hold off;
%subplot(222);plot(1./Va,im3,'-');
subplot(223);plot(1./Va,D,'-');
subplot(224);plot(1./Va,G,'-');

subplot(222);
for i=0:10
n=i*19+1;
d=D(n);g=G(n);a=Va(n);
E=exp((z-g)/d);
y=(a/d)*E./((1+E).^(a+1));
hold on;plot(z,y);hold off;
end
subplot(222);
for i=0:10
n=i*19+1;
d=Di(n);g=Gi(n);a=1/Va(n);
E=exp((z-g)/d);
y=(a/d)*E./((1+E).^(a+1));
hold on;plot(z,y,'-');hold off;
end

figure(2);clf;
subplot(221);plot(1./Va,im3,'-');
hold on;plot(1./Va,y2,'r');hold off;
%subplot(222);plot(1./Va,im3,'-');
subplot(223);plot(1./Va,Di,'-');
subplot(224);plot(1./Va,Gi,'-');

subplot(222);
for i=0:10
n=i*19+1;
d=D(n);g=G(n);a=Va(n);
E=exp((-g-z)/d);
y=(a/d)*E./((1+E).^(a+1));
hold on;plot(z,y);hold off;
end
subplot(222);
for i=0:10
n=i*19+1;
d=Di(n);g=Gi(n);a=1/Va(n);
E=exp((-g-z)/d);
y=(a/d)*E./((1+E).^(a+1));
hold on;plot(z,y,'-');hold off;
end

```

B.2.4 Polynomial objective function for for HG0 parameters estimation

```
function y = f1(x,m3)

p1=[-0.1612    0.2727    1.0277   -1.1390];
p2=[4.6168,-16.2866,21.8900,-12.4171,0.2030,1.9943];
if abs(m3)<1.08
y=m3-polyval(p1,x);
else
y=abs(m3)-polyval(p2,x);
end
```

B.2.5 Parameters Estimation

```
function [Pt,par]= Hg0p1(z)

m3=mean(z.^3);
fun=@(x) f1(x,m3);
a=fzero(fun,0.5);

if m3<0
a=1/a;
end

d=1/sqrt(psi(1,1)+psi(1,a));
g=d*(psi(1)-psi(a));

nr=rand(100);
nr=nr(:);
Pt=-sign(m3)*(d*log((nr.^(-1/a))-1)+g);
mu=mean(Pt);sg=std(Pt);
Pt=(Pt-mu)/sg;

par=[a,d,g];
```

B.2.6 Kullback-Leibler divergence

Developed by Nima Razavi

<http://es.mathworks.com/matlabcentral/fileexchange/20688-kullback-leibler-divergence>

```
function dist=KLDiv(P,Q)
% dist = KLDiv(P,Q) Kullback-Leibler divergence of two
discrete
% probability distributions
% P and Q are automatically normalised to have the sum
of one on rows
% have the length of one at each
% P = n x nbins
% Q = 1 x nbins or n x nbins(one to one)
% dist = n x 1

if size(P,2)~=size(Q,2)
error('the number of columns in P and Q should be the same');
end

if sum(~isfinite(P(:))) + sum(~isfinite(Q(:)))
error('the inputs contain non-finite values!')
end

% normalizing the P and Q
if size(Q,1)==1
Q = Q ./sum(Q);
P = P ./repmat(sum(P,2),[1 size(P,2)]);
temp = P.*log(P./repmat(Q,[size(P,1) 1]));
temp(isnan(temp))=0;% resolving the case when P(i)==0
dist = sum(temp,2);

elseif size(Q,1)==size(P,1)

Q = Q ./repmat(sum(Q,2),[1 size(Q,2)]);
P = P ./repmat(sum(P,2),[1 size(P,2)]);
temp = P.*log(P./Q);
temp(isnan(temp))=0; % resolving the case when P(i)==0
dist = sum(temp,2);
end
```

Appendix C

MATLAB Code for Stable Distributions

The code for Stable Distribution was developed by by Mark Veillette
<http://es.mathworks.com/matlabcentral/fileexchange/37514-stbl-alpha-stable-distributions-for-matlab>
This open source version was downloaded on August 23 2010

C.1 Stable Distribution PDF

```
function p = stblpdf(x,alpha,beta,gam,delta,varargin)
%P = STBLPDF(X,ALPHA,BETA,GAM,DELTA) returns the pdf of the stable
% distribtuion with characteristic exponent ALPHA, skewness BETA,
% scale parameter GAM, and location parameter DELTA, at the values
% in X. We use the parameterization of stable distribtuions used in
% [2] - The characteristic function phi(t) of a
% S(ALPHA,BETA,GAM,DELTA) random variable has the form
%
% phi(t) = exp(-GAM^ALPHA |t|^ALPHA
% [1 - i BETA (tan(pi ALPHA/2) sign(t)]+ i DELTA t )
% if alpha ~= 1
% phi(t) = exp(-GAM |t| [ 1 + i BETA (2/pi) (sign(t)) log|t|]
% + i DELTA t
% if alpha = 1
% The size of P is the size of X. ALPHA,BETA,GAM and DELTA must
% be scalars
%P = STBLPDF(X,ALPHA,BETA,GAM,DELTA,TOL) computes the pdf
% to within an absolute error of TOL.
%
% The algorithm works by computing the numerical integrals in
% Theorem1 in [1] using MATLAB's QUADV function. The integrands
% are smooth non-negative functions, but for certain parameter
% values
% can have sharp peaks which might be missed. To avoid this,
STBLEPDF
% locates the maximum of this integrand and breaks the integral
into two
% pieces centered around this maximum (this is exactly
% the idea suggested in [1] ).
%
```

```

% If abs(ALPHA - 1) < 1e-5, ALPHA is rounded to 1.
%
%P = STBLPDF(...,'quick') skips the step of locating the peak
in the
% integrand, and thus is faster, but is less accurate deep
into the tails
% of the pdf. This option is useful for plotting. In place
of 'quick',
% STBLPDF also accepts a logical true or false
(for quick or not quick)
%
% See also: STBLRND, STBLCDF, STBLINV, STBLFIT
%
% References:
%
% [1] J. P. Nolan (1997)
%      "Numerical Calculation of Stable Densities
and Distribution
%      Functions" Commun. Statist. - Stochastic Modles,
13(4), 759-774
%
% [2] G Samorodnitsky, MS Taqqu (1994)
%      "Stable non-Gaussian random processes: stochastic
models with
%      infinite variance" CRC Press
%

if nargin < 5
error('stblpdf:TooFewInputs','Requires at least five
input arguments.');
```

```

end

% Check parameters
if alpha <= 0 || alpha > 2 || ~isscalar(alpha)
error('stblpdf:BadInputs',' "alpha" must be a scalar which
lies in the interval (0,2]');
```

```

end
if abs(beta) > 1 || ~isscalar(beta)
error('stblpdf:BadInputs',' "beta" must be a scalar which lies
in the interval [-1,1]');
```

```

end
if gam < 0 || ~isscalar(gam)
error('stblpdf:BadInputs',' "gam" must be a non-negative scalar');
```

```

end
if ~isscalar(delta)
error('stblpdf:BadInputs',' "delta" must be a scalar');
```

```

end

% Warn if alpha is very close to 1 or 0
if ( 1e-5 < abs(1 - alpha) && abs(1 - alpha) < .02) || alpha < .02
```



```

warning('stblpdf:ScaryAlpha',...
'Difficult to approximate pdf for alpha close to 0 or 1')
end

% warnings will happen during call to QUADV, and it's okay
warning('off');

% Check and initialize additional inputs
quick = false;
tol = [];
for i=1:length(varargin)
if strcmp(varargin{i},'quick')
quick = true;
elseif islogical(varargin{i})
quick = varargin{end};
elseif isscalar(varargin{i})
tol = varargin{i};
end
end

if isempty(tol)
if quick
tol = 1e-8;
else
tol = 1e-12;
end
end

%===== Compute pdf =====%

% Check to see if you are in a simple case, if so be quick,
% if not do general algorithm
if alpha == 2 % Gaussian distribution
x = (x - delta)/gam; % Standardize
p = 1/sqrt(4*pi) * exp( -.25 * x.^2 ); % ~ N(0,2)
p = p/gam; %rescale

elseif alpha==1 && beta == 0 % Cauchy distribution
x = (x - delta)/gam; % Standardize
p = (1/pi) * 1./(1 + x.^2);
p = p/gam; %rescale

elseif alpha == .5 && abs(beta) == 1 % Levy distribution
x = (x - delta)/gam; % Standardize
p = zeros(size(x));
if beta ==1
p( x <= 0 ) = 0;
p( x > 0 ) = sqrt(1/(2*pi)) * exp(-.5./x(x>0)) ./...
x(x>0).^1.5;

```

```

else
p(x >= 0) = 0;
p(x < 0) = sqrt(1/(2*pi)) * exp(.5./x(x<0) ) ./...
( -x(x<0) ).^1.5;
end
p = p/gam; %rescale

elseif abs(alpha - 1) > 1e-5 % Gen. Case, alpha ~= 1

xold = x; % Save for later
% Standardize in (M) parameterization ( See equation (2) in [1] )
x = (x - delta)/gam - beta * tan(alpha*pi/2);

% Compute pdf
p = zeros(size(x));
zeta = - beta * tan(pi*alpha/2);
theta0 = (1/alpha) * atan(beta*tan(pi*alpha/2));
A1 = alpha*theta0;
A2 = cos(A1)^(1/(alpha-1));
expl = alpha/(alpha-1);
alham1 = alpha - 1;
c2 = alpha ./ (pi * abs(alpha - 1) * ( x(x>zeta) - zeta) );
V = @(theta) A2 * ( cos(theta) ./
sin( alpha*(theta + theta0) ) ).^expl.*...
cos( A1 + alham1*theta ) ./ cos(theta);

% x > zeta, calculate integral using QUADV
if any(x > zeta)
xshift = (x(x>zeta) - zeta) .^ expl;

if beta == -1 && alpha < 1
p(x > zeta) = 0;
elseif ~quick % Locate peak in integrand and split up integral
g = @(theta) xshift(:) .* V(theta) - 1;
R = repmat([-theta0, pi/2 ],numel(xshift),1);
if abs(beta) < 1
theta2 = bisectionSolver(g,R,alpha);
else
theta2 = bisectionSolver(g,R,alpha,beta,xshift);
end
theta2 = reshape(theta2,size(xshift));
% change variables so the two integrals go from
% 0 to 1/2 and 1/2 to 1.
theta2shift1 = 2*(theta2 + theta0);
theta2shift2 = 2*(pi/2 - theta2);
g1 = @(theta) xshift .* ...
V(theta2shift1 * theta - theta0);
g2 = @(theta) xshift .* ...
V(theta2shift2 * (theta - .5) + theta2);

```

```

zexpz = @(z) max(0,z .* exp(-z)); % use max incase of NaN

p(x > zeta) = c2 .* ...
(theta2shift1 .* quadv(@(theta) zexpz( g1(theta) ),...
0 , .5, tol) ...
+ theta2shift2 .* quadv(@(theta) zexpz( g2(theta) ),...
.5 , 1, tol) );

else % be quick - calculate integral without locating peak
% Use a default tolerance of 1e-6
g = @(theta) xshift * V(theta);
zexpz = @(z) max(0,z .* exp(-z)); % use max incase of NaN
p( x > zeta ) = c2 .* quadv(@(theta) zexpz( g(theta) ),...
-theta0 , pi/2, tol );
end
p(x > zeta) = p(x>zeta)/gam; %rescale

end

% x = zeta, this is easy
if any( abs(x - zeta) < 1e-8 )
p( abs(x - zeta) < 1e-8 ) = max(0,gamma(1 + 1/alpha)*...
cos(theta0)/(pi*(1 + zeta^2)^(1/(2*alpha)))));
p( abs(x - zeta) < 1e-8 ) = p( abs(x - zeta) < 1e-8 )/gam;
%rescale

end

% x < zeta, recall function with -xold, -beta, -delta
% This doesn't need to be rescaled.
if any(x < zeta)
p( x < zeta ) = stblpdf( -xold( x<zeta ),alpha,-beta,...
gam , -delta , tol , quick);
end

else % Gen case, alpha = 1

x = (x - (2/pi) * beta * gam * log(gam) - delta)/gam;
% Standardize

% Compute pdf
piover2 = pi/2;
twooverpi = 2/pi;
oneoverb = 1/beta;
theta0 = piover2;
% Use logs to avoid overflow/underflow
logV = @(theta) log(twooverpi * ((piover2 + beta *theta)./
cos(theta))) + ...
( oneoverb * (piover2 + beta *theta) .* tan(theta) );
c2 = 1/(2*abs(beta));

```

```

xterm = ( -pi*x/(2*beta));

if ~quick % Locate peak in integrand and split up integral
% Use a default tolerance of 1e-12
logg = @(theta) xterm(:) + logV(theta) ;
R = repmat([-theta0, pi/2 ],numel(xterm),1);
theta2 = bisectionSolver(logg,R,1-beta);
theta2 = reshape(theta2,size(xterm));
% change variables so the two integrals go from
% 0 to 1/2 and 1/2 to 1.
theta2shift1 = 2*(theta2 + theta0);
theta2shift2 = 2*(pi/2 - theta2);
logg1 = @(theta) xterm + ...
logV(theta2shift1 * theta - theta0);
logg2 = @(theta) xterm + ...
logV(theta2shift2 * (theta - .5) + theta2);
zexpz = @(z) max(0,exp(z) .* exp(-exp(z)));
% use max incase of NaN

p = c2 .* ...
(theta2shift1 .* quadv(@(theta) zexpz( logg1(theta) ),...
0 , .5, tol) ...
+ theta2shift2 .* quadv(@(theta) zexpz( logg2(theta) ),...
.5 , 1, tol) );

else % be quick - calculate integral without locating peak
% Use a default tolerance of 1e-6
logg = @(theta) xterm + logV(theta);
zexpz = @(z) max(0,exp(z) .* exp(-exp(z)));
% use max incase of NaN
p = c2 .* quadv(@(theta) zexpz( logg(theta) ),...
-theta0 , pi/2, tol );

end

p = p/gam; %rescale

end

p = real(p); % just in case a small imaginary piece crept in
% This might happen when (x - zeta) is really small

end

function X = bisectionSolver(f,R,alpha,varargin)
% Solves equation g(theta) - 1 = 0 in STBLPDF using a

```

```

%vectorized bisection
% method and a tolerance of 1e-5. The solution to this
% equation is used to increase accuracy in the calculation
% of a numerical integral.
%
% If alpha ~= 1 and 0 <= beta < 1, the equation always has
% a solution
% If alpha > 1 and beta <= 1, then g is monotone decreasing
%
% If alpha < 1 and beta < 1, then g is monotone increasing
%
% If alpha = 1, g is monotone increasing if beta > 0 and monotone
% decreasing if beta < 0. Input alpha = 1 - beta
% to get desired results.
%

if nargin < 2
error('bisectionSolver:TooFewInputs','Requires at least two input
arguments.');
```

```

end

noSolution = false(size(R,1));
% if ~isempty(varargin)
%     beta = varargin{1};
%     xshift = varargin{2};
%     if abs(beta) == 1
%         V0=(1/alpha)^(alpha/(alpha-1))*(1-alpha)*...
%         cos(alpha*pi/2)*xshift;
%         if alpha > 1
%             noSolution = V0 - 1 %>= 0;
%         elseif alpha < 1
%             noSolution = V0 - 1 %<= 0;
%         end
%     end
% end

tol = 1e-6;
maxiter = 30;

[N M] = size(R);
if M ~= 2
error('bisectionSolver:BadInput',...
'"R" must have 2 columns');
end

a = R(:,1);
b = R(:,2);
X = (a+b)/2;

```

```

try
val = f(X);
catch ME
error('bisectionSolver:BadInput',...
'Input function inconsistant with rectangle dimension')
end

if size(val,1) ~= N
error('bisectionSolver:BadInput',...
'Output of function must be a column vector
with dimension of input');
end

% Main loop
val = inf;
iter = 0;

while( max(abs(val)) > tol && iter < maxiter )
X = (a + b)/2;
val = f(X);
l = (val > 0);
if alpha > 1
l = 1-l;
end
a = a.*l + X.*(1-l);
b = X.*l + b.*(1-l);
iter = iter + 1;
end

if any(noSolution)
X(noSolution) = (R(1,1) + R(1,2))/2;
end

end

```

C.2 Stable Distribution CDF

```

function F = stblcdf(x,alpha,beta,gam,delta,varargin)
%F = STBLCDF(X,ALPHA,BETA,GAM,DELTA) returns the cdf of the stable
% distribtuion with characteristic exponent ALPHA, skewness BETA,
scale
% parameter GAM, and location parameter DELTA, at the values in X.
We use
% the parameterization of stable distribtuions used in [2] -
The

```

```

% characteristic function phi(t) of a S(ALPHA,BETA,GAM,DELTA)
% random variable has the form
%
% phi(t) = exp(-GAM^ALPHA |t|^ALPHA [1 - i BETA (tan(pi ALPHA/2)
sign(t)]
%
%           + i DELTA t )   if alpha ~= 1
%
% phi(t) = exp(-GAM |t| [ 1 + i BETA (2/pi) (sign(t)) log|t| ] + i
DELTA t
%
%           if alpha = 1
%
% The size of P is the size of X.  ALPHA,BETA,GAM and DELTA must
be scalars
%
% P = STBLCDF(X,ALPHA,BETA,GAM,DELTA,TOL) computes the cdf to within
an
% absolute error of TOL.  Default for TOL is 1e-8.
%
% The algorithm works by computing the numerical integrals in
Theorem
% 1 in [1] using MATLAB's QUADV function.
%
% If abs(ALPHA - 1) < 1e-5,  ALPHA is rounded to 1.
%
% See also: STBLRND, STBLPDF, STBLINV
%
% References:
%
% [1] J. P. Nolan (1997)
%     "Numerical Calculation of Stable Densities and Distribution
%     Functions"  Commun. Statist. - Stochastic Modles, 13(4),
759-774
%
% [2] G Samorodnitsky, MS Taqqu (1994)
%     "Stable non-Gaussian random processes: stochastic models with
%     infinite variance"  CRC Press
%
%
%
%
if nargin < 5
error('stblcdf:TooFewInputs','Requires at least five input arguments.');
```

```

end

% Check parameters
if alpha <= 0 || alpha > 2 || ~isscalar(alpha)
error('stblcdf:BadInputs',' "alpha" must be a scalar which lies in the
interval (0,2]');
```

```

end
if abs(beta) > 1 || ~isscalar(beta)

```

```

error('stblcdf:BadInputs',' "beta" must be a scalar which lies in the
interval [-1,1]');
end
if gam < 0 || ~isscalar(gam)
error('stblcdf:BadInputs',' "gam" must be a non-negative scalar');
end
if ~isscalar(delta)
error('stblcdf:BadInputs',' "delta" must be a scalar');
end

if nargin > 6
error('stblcdf:TooManyInputs','Accepts at most six input arguments. ');
elseif isempty(varargin)
tol = 1e-8;
elseif isscalar(varargin{1})
tol = varargin{1};
else
error('stblcdf:BadInput',' "TOL" must be a scalar. ')
end

% Warn if alpha is very close to 1 or 0
if (1e-5 < abs(1 - alpha) && abs(1 - alpha) < .02) || alpha < .02
warning('stblcdf:ScaryAlpha',...
'Difficult to approximate cdf for alpha close to 0 or 1')
end

%===== Compute CDF =====%

% Check to see if you are in a simple case, if so be quick, if not do
% general algorithm
if alpha == 2 % Gaussian distribution
x = (x - delta)/gam; % Standardize
F = .5*(1 + erf(x/2)); % ~ N(0,2)

elseif alpha==1 && beta == 0 % Cauchy distribution
x = (x - delta)/gam; % Standardize
F = 1/pi * atan(x) + .5;

elseif alpha == .5 && abs(beta) == 1 % Levy distribution
x = (x - delta)/gam; % Standardize
F = zeros(size(x));
if beta > 0
F(x > 0) = erfc(sqrt(1./(2*x(x>0))));
F(x <= 0) = 0;
else % beta < 0
F(x < 0) = 1 - erfc(sqrt(-1./(2*x(x<0))));
F(x >= 0) = 1;
end

elseif abs(alpha - 1) > 1e-5 % Gen. Case, alpha ~= 1

```



```

xold = x; % Save for possible later use
% Standardize in (M) parameterization ( See equation (2) in [1] )
x = (x - delta)/gam - beta * tan(alpha*pi/2);
F = zeros(size(x));
% Compute CDF
zeta = -beta * tan(pi*alpha/2);
theta0 = (1/alpha) * atan( beta * tan(pi*alpha/2) );
A1 = alpha*theta0;
c1 = (alpha > 1) + (alpha < 1)*(1/pi)*(pi/2 - theta0);
A2 = cos(A1)^(1/(alpha-1));
expl = alpha/(alpha-1);
alphan1 = alpha - 1;
V = @(theta) A2 * ( cos(theta) ./ sin( alpha*(theta + theta0) ) )
.^expl.*...
cos( A1 + alphan1*theta ) ./ cos(theta);

if any(x > zeta)
xshift = (x(x>zeta) - zeta).^(alpha/(alpha - 1));
% shave off end points of integral to avoid numerical instability
% when calculating V
F( x > zeta ) = c1 + sign(1-alpha)/pi * ...
quadv(@(theta) exp(-xshift * V(theta)),...
-theta0+1e-10,pi/2-1e-10,tol);
end

if any(abs(x - zeta) < 1e-8)
F(abs(x - zeta) < 1e-8) = (1/pi) * (pi/2 - theta0);
end

if any( x < zeta)
% Recall with -xold, -beta, -delta
F(x < zeta) =...
1 - stblcdf(-xold(x < zeta),alpha,-beta,gam,-delta);
end
elseif beta > 0
% Gen. Case, alpha = 1, beta >0
x = (x - (2/pi) * beta * gam * log(gam) - delta)/gam;
% Standardize
piover2 = pi/2;
twooverpi = 2/pi;
oneoverb = 1/beta;
% Use logs to avoid overflow/underflow
logV = @(theta) log(twooverpi * ((piover2 + beta *theta)
./cos(theta))) + ...
( oneoverb * (piover2 + beta *theta) .* tan(theta) );

xterm = (-pi*x/(2*beta));
F = (1/pi)*quadv(@(theta) exp(-exp(xterm + logV(theta))),...

```

```

-pi/2+1e-12,pi/2-1e-12,tol);

else                                     % alpha = 1, beta < 0
F = 1 - stblcdf(-x,1,-beta,gam,-delta,tol);
end

F = max(real(F),0);
% in case of small imaginary or negative results from QUADV
end

```

C.3 Stable Distributed Random Numbers

```

function r = stblrnd(alpha,beta,gamma,delta,varargin)
%STBLRND alpha-stable random number generator.
% R = STBLRND(ALPHA,BETA,GAMMA,DELTA) draws a sample
from the Levy
% alpha-stable distribution with characteristic
exponent ALPHA,
% skewness BETA, scale parameter GAMMA and location
parameter DELTA.
% ALPHA,BETA,GAMMA and DELTA must be scalars which
fall in the following
% ranges :
%   0 < ALPHA <= 2
%   -1 <= BETA <= 1
%   0 < GAMMA < inf
%   -inf < DELTA < inf
%
%
% R = STBLRND(ALPHA,BETA,GAMMA,DELTA,M,N,...) or
% R = STBLRND(ALPHA,BETA,GAMMA,DELTA,[M,N,...])
returns an M-by-N-by-...
% array.
%
%
% References:
% [1] J.M. Chambers, C.L. Mallows and B.W. Stuck (1976)
%     "A Method for Simulating Stable Random Variables"
%     JASA, Vol. 71, No. 354. pages 340-344
%
% [2] Aleksander Weron and Rafal Weron (1995)
%     "Computer Simulation of Levy alpha-Stable Variables
and Processes"
%     Lec. Notes in Physics, 457, pages 379-392
%

```

```

if nargin < 4
error('stats:stblrnd:TooFewInputs','Requires at least four
input arguments.');
```

```

end

% Check parameters
if alpha <= 0 || alpha > 2 || ~isscalar(alpha)
error('stats:stblrnd:BadInputs',' "alpha" must be a scalar
which lies in the interval (0,2]');
```

```

end
if abs(beta) > 1 || ~isscalar(beta)
error('stats:stblrnd:BadInputs',' "beta" must be a scalar
which lies in the interval [-1,1]');
```

```

end
if gamma < 0 || ~isscalar(gamma)
error('stats:stblrnd:BadInputs',' "gamma" must be a
non-negative scalar');
```

```

end
if ~isscalar(delta)
error('stats:stblrnd:BadInputs',' "delta" must be
a scalar');
```

```

end

% Get output size
[err, sizeOut] = genOutsize(4,alpha,beta,gamma,delta,
varargin{:});
if err > 0
error('stats:stblrnd:InputSizeMismatch','Size information
is inconsistent.');
```

```

end

```

C.4 Stable Distribution data fitting

```

function params = stblfit(X,varargin)
%PARAMS = STBLFIT(X) returns an estimate of the
four parameters in a
% fit of the alpha-stable distribution to the data X.
The output
% PARAMS is a 4 by 1 vector which contains the estimates of the
% characteristic exponent ALPHA, the skewness BETA,
the scale GAMMA and
% location DELTA.
%
%PARAMS = STBLFIT(X,METHOD) Specifies the algorithm used to
% estimate the parameters. The choices for METHOD are

```

```

%      'ecf' - Fits the four parameters to the empirical
characteristic
%      function estimated from the data.
This is the default.
%      Based on Koutrouvelis (1980,1981),
see [1],[2] below.
%      'percentile' - Fits the four parameters using various
%      percentiles of the data X.
This is faster than 'ecf',
%      however studies have shown it
to be slightly less
%      accurate in general.
%      Based on McCulloch (1986),
see [2] below.
%
%PARAMS = STBLFIT(...,OPTIONS) specifies options used in STBLFIT.
OPTIONS
% must be an options stucture created with the STATSET function.
Possible
% options are
%      'Display' - When set to 'iter', will display the values
of
%      alpha,beta,gamma and delta in each
%      iteration. Default is 'off'.
%      'MaxIter' - Specifies the maximum number of iterations
allowed in
%      estimation. Default is 5.
%      'TolX' - Specifies threshold to stop iterations.
Default is
%      0.01.
%
% See also: STBLRND, STBLPDF, STBLCDF, STBLINV
%
% References:
% [1] I. A. Koutrouvelis (1980)
%      "Regression-Type Estimation of the Paramters of
Stable Laws.
%      JASA, Vol 75, No. 372
%
% [2] I. A. Koutrouvelis (1981)
%      "An Iterative Procedure for the estimation of the
Parameters of
%      Stable Laws"
%      Commun. Stat. - Simul. Comput. 10(1), pages 17-28
%
% [3] J. H. McCulloch (1986)
%      "Simple Consistent Estimators of Stable Distribution
Parameters"
%      Cummun. Stat. Simul. Comput. 15(4)
%

```

```

% [4] A. H. Welsh (1986)
% "Implementing Empirical Characteristic Function
Procedures"
% Statistics & Probability Letters Vol 4, pages 65-67

% ==== Gather additional options
dispit = false;
maxiter = 5;
tol = .01;
if ~isempty(varargin)
if isstruct(varargin{end})
opt = varargin{end};
try
dispit = opt.Display;
catch ME
error('OPTIONS must be a structure created
with STATSET');
end
if ~isempty(opt.MaxIter)
maxiter = opt.MaxIter;
end
if ~isempty(opt.TolX)
tol = opt.TolX;
end
end
end

if strcmp(dispit,'iter')
dispit = true;
fprintf('      iteration\t      alpha\t      beta\t
gamma\t\t
delta\n');
dispfmt = '%8d\t%14g\t%8g\t%8g\t%8g\n';
end

% == Find which method.
if any(strcmp(varargin,'percentile'))
maxiter = 0; % This is McCulloch's percentile method
end

% ==== Begin estimation =====
N = numel(X); % data size

% function handle to compute empirical char. functions
I = sqrt(-1);
phi = @(theta,data) 1/numel(data) * sum( exp( I * ...
reshape(theta,numel(theta),1) *...
reshape(data,1,numel(data)) ) , 2);

```

```

% Step 1 - Obtain initial estimates of parameters
using McCulloch's method
%           then standardize data
[alpha beta] = intAlpBet(X);
[gam delta ] = intGamDel(X,alpha,beta);

if gam==0
% Use standard deviation as initial guess
gam = std(X);
end

s = (X - delta)/gam;

if dispit
fprintf(dispfmt,0,alpha,beta,gam,delta);
end

% Step 2 - Iterate until convergence
alphaold = alpha;
deltaold = delta;
diffbest = inf;
for iter = 1:maxiter

% Step 2.1 - Regress against ecf to refine estimates
of alpha & gam
%           After iteration 1, use generalized
least squares
if iter <= 2
K = chooseK(alpha,N);
t = (1:K)*pi/25;
w = log(abs(t));
end

y = log( - log( abs(phi(t,s)).^2 ) );

if iter == 1 % use ordinary least squares regression
ell = regress(y,[w' ones(size(y))]);
alpha = ell(1);
gamhat = (exp(ell(2))/2)^(1/alpha);
gam = gam * gamhat;
else % use weighted least squares regression
sig = charCov1(t ,N, alpha , beta, 1);
try
ell = lscov([w' ones(size(y))],y,sig);
catch % In case of badly conditioned covariance matrix,
just use diagonal entries
try
ell = lscov([w' ones(size(y))],y,eye(K).*(sig+eps));

```

```

catch
break
end
end
alpha = ell(1);
gamhat = (exp(ell(2))/2)^(1/alpha);
gam = gam * gamhat;
end

% Step 2.2 - Rescale data by estimated scale, truncate
s = s/gamhat;
alpha = max(alpha,0);
alpha = min(alpha,2);
beta = min(beta,1);
beta = max(beta,-1);
gam = max(gam,0);

% Step 2.3 - Regress against ecf to refine estimates
of beta,
delta
%           After iteration 1, use generalized least
squares
if iter <= 2
L = chooseL(alpha,N);
% To ensure g is continuous, find first zero in real
part of ecf
A = efcRoot(s);
u = (1:L)*min(pi/50,A/L);
end

ecf = phi(u,s);
U = real(ecf);
V = imag(ecf);
g = atan2(V,U);
if iter == 1 % use ordinary least squares
ell = regress(g, [u', sign(u').*abs(u').^alpha]);
beta = ell(2)/tan(alpha*pi/2) ;
delta = delta + gam* ell(1) ;
else % use weighted least squares regression
sig = charCov2(u ,N, alpha , beta, 1);
try
ell = lscov([u', sign(u').*abs(u').^alpha],g,sig);
catch % In case of badly conditioned covariance matrix,
use diagonal entries
try
ell = lscov([u', sign(u').*abs(u').^alpha],g,eye(L).*(sig+eps));
catch
break
end
end

```

```
end
beta = ell(2)/tan(alpha*pi/2) ;
delta = delta + gam* ell(1) ;
end

% Step 2.4 Remove estimated shift
s = s - ell(1);

% display
if dispit
fprintf(dispfmt,iter,alpha,beta,gam,delta);
end

% Check for blow-up
if any(isnan([alpha, beta, gam, delta]) |
isinf([alpha, beta, gam, delta]))
break
end

% Step 2.5 Check for convergence, keep track of
parameters with
% smallest 'diff'
diff = (alpha - alphaold)^2 + (delta - deltaold)^2;
if abs(diff) < diffbest
bestparams = [alpha; beta; gam; delta];
diffbest = diff;
if diff < tol
break;
end
end

alphaold = alpha;
deltaold = delta;

end

% Pick best
if maxiter > 0 && iter >= 1
alpha = bestparams(1);
beta = bestparams(2);
gam = bestparams(3);
delta = bestparams(4);
end

% Step 3 - Truncate if necessary
alpha = max(alpha,0);
```



```

alpha = min(alpha,2);
beta = min(beta,1);
beta = max(beta,-1);
gam = max(gam,0);

params = [alpha; beta; gam; delta];

end % End stblfit

%=====
%=====

function [alpha beta] = intAlpBet(X)
% Interpolates Tables found in MuCulloch (1986) to obtain a
starting
% estimate of alpha and beta based on percentiles of data X

% Input tables
nuA = [2.439 2.5 2.6 2.7 2.8 3.0 3.2 3.5 4.0 5.0 6.0 8.0
10 15 25];
nuB = [0 .1 .2 .3 .5 .7 1];
[a b] = meshgrid( nuA , nuB );
alphaTab= [2.000 2.000 2.000 2.000 2.000 2.000 2.000;...
1.916 1.924 1.924 1.924 1.924 1.924 1.924;...
1.808 1.813 1.829 1.829 1.829 1.829 1.829;...
1.729 1.730 1.737 1.745 1.745 1.745 1.745;...
1.664 1.663 1.663 1.668 1.676 1.676 1.676;...
1.563 1.560 1.553 1.548 1.547 1.547 1.547;...
1.484 1.480 1.471 1.460 1.448 1.438 1.438;...
1.391 1.386 1.378 1.364 1.337 1.318 1.318;...
1.279 1.273 1.266 1.250 1.210 1.184 1.150;...
1.128 1.121 1.114 1.101 1.067 1.027 0.973;...
1.029 1.021 1.014 1.004 0.974 0.935 0.874;...
0.896 0.892 0.887 0.883 0.855 0.823 0.769;...
0.818 0.812 0.806 0.801 0.780 0.756 0.691;...
0.698 0.695 0.692 0.689 0.676 0.656 0.595;...
0.593 0.590 0.588 0.586 0.579 0.563 0.513]';
betaTab= [ 0.000 2.160 1.000 1.000 1.000 1.000 1.000;...
0.000 1.592 3.390 1.000 1.000 1.000 1.000;...
0.000 0.759 1.800 1.000 1.000 1.000 1.000;...
0.000 0.482 1.048 1.694 1.000 1.000 1.000;...
0.000 0.360 0.760 1.232 2.229 1.000 1.000;...
0.000 0.253 0.518 0.823 1.575 1.000 1.000;...
0.000 0.203 0.410 0.632 1.244 1.906 1.000;...
0.000 0.165 0.332 0.499 0.943 1.560 1.000;...
0.000 0.136 0.271 0.404 0.689 1.230 2.195;...
0.000 0.109 0.216 0.323 0.539 0.827 1.917;...
0.000 0.096 0.190 0.284 0.472 0.693 1.759;...
0.000 0.082 0.163 0.243 0.412 0.601 1.596;...
0.000 0.074 0.147 0.220 0.377 0.546 1.482;...

```

```

0.000 0.064 0.128 0.191 0.330 0.478 1.362;...
0.000 0.056 0.112 0.167 0.285 0.428 1.274]';

% Calculate percentiles
Xpcts = prctile(X,[95 75 50 25 5]);
nuAlpha = (Xpcts(1) - Xpcts(5))/(Xpcts(2) - Xpcts(4));
nuBeta = (Xpcts(1) + Xpcts(5) - 2*Xpcts(3))/(Xpcts(1) -
Xpcts(5));
% Bring into range
if nuAlpha < 2.4390
nuAlpha = 2.439 + 1e-12;
elseif nuAlpha > 25
nuAlpha = 25 - 1e-12;
end

s = sign(nuBeta);

% Get alpha
alpha = interp2(a,b,alphaTab,nuAlpha,abs(nuBeta));

% Get beta
beta = s * interp2(a,b,betaTab,nuAlpha,abs(nuBeta));

% Truncate beta if necessary
if beta>1
beta = 1;
elseif beta < -1
beta = -1;
end

end

function [gam delta] = intGamDel(X,alpha,beta)
% Uses McCulloch's Method to obtain scale and location
of data X given
% estimates of alpha and beta.

% Get percentiles of data and true percentiles given
alpha and beta;
Xpcts = prctile(X,[75 50 25]);

% If alpha is very close to 1, truncate to avoid numerical
instability.
warning('off','stblcdf:ScaryAlpha');
warning('off','stblpdf:ScaryAlpha');
if abs(alpha - 1) < .02
alpha = 1;
end

```

```

% With the 'quick' option, these are equivalent to
McCulloch's tables
Xquart = stblinv([.75 .25],alpha,beta,1,0,'quick');
Xmed = stblinv(.5,alpha,beta,1,-beta*tan(pi*alpha/2),'quick');

% Obtain gamma as ratio of interquartile ranges
gam = (Xpcts(1) - Xpcts(3))/(Xquart(1) - Xquart(2));

% Obtain delta using median of shifted data
and estimate of gamma
zeta = Xpcts(2) - gam * Xmed;
delta = zeta - beta*gam*tan(alpha*pi/2);

end

function K = chooseK(alpha,N)
% Interpolates Table 1 in [1] to calculate optimum K
given alpha and N

% begin parameters into correct ranges.
alpha = max(alpha,.3);
alpha = min(alpha,1.9);
N = max(N,200);
N = min(N,1600);
a = [1.9, 1.5: -.2: .3];
n = [200 800 1600];
[X Y] = meshgrid(a,n);
Kmat = [ 9   9   9 ; ...
11  11  11 ; ...
22  16  14 ; ...
24  18  15 ; ...
28  22  18 ; ...
30  24  20 ; ...
86  68  56 ; ...
134 124 118 ];
K = round(interp2(X,Y,Kmat',alpha,N,'linear'));

end

function L = chooseL(alpha,N)
% Interpolates Table 2 in [1] to calculate optimum L given
alpha and N

alpha = max(alpha,.3);
alpha = min(alpha,1.9);
N = max(N,200);
N = min(N,1600);
a = [1.9, 1.5, 1.1:-.2:.3];
n = [200 800 1600];

```

```

[X Y] = meshgrid(a,n);
Lmat = [ 9  10  11 ; ...
12  14  15 ; ...
16  18  17 ; ...
14  14  14 ; ...
24  16  16 ; ...
40  38  36 ; ...
70  68  66 ];
L = round(interp2(X,Y,Lmat',alpha,N,'linear'));

end

function A = efcRoot(X)
% An iterative procedure to find the first positive
root of the real part
% of the empirical characteristic function of the data X.
Based on [4].

N = numel(X);
U = @(theta) 1/N * sum( cos( ...
reshape(theta,numel(theta),1) *...
reshape(X,1,N) ) , 2 ); % Real part of ecf
m = mean(abs(X));
A = 0;
val = U(A);
iter1 = 0;
while abs(val) > 1e-3 && iter1 < 10^4
A = A + val/m;
val = U(A);
iter1 = iter1 + 1;
end

end

function sig = charCov1(t ,N, alpha , beta,gam)
% Compute covariance matrix of  $y = \log(-\log(\phi(t)))$ ,
where  $\phi(t)$  is
% ecf of alpha-stable random variables. Based on
Theorem in [2].

K = length(t);
w = tan(alpha*pi/2);
calpha = gam^alpha;

Tj = repmat( t(:) , 1 , K);
Tk = repmat( t(:)' , K , 1);
Tjalpha = abs(Tj).^alpha;
Tkalpha = abs(Tk).^alpha;
TjxTk = abs(Tj .* Tk);
TjpTk = Tj + Tk ;

```

```

TjpTkalpha = abs(TjpTk).^alpha;
TjmTk = Tj - Tk ;
TjmTkalpha = abs(TjmTk).^alpha;

A = calpha*( Tjalpha + Tkalpha - TjmTkalpha);
B = calpha * beta *...
(-Tjalpha .* sign(Tj) * w ...
+ Tkalpha .* sign(Tk) * w ...
+ TjmTkalpha .* sign(TjmTk) * w) ;
D = calpha * (Tjalpha + Tkalpha - TjpTkalpha);
E = calpha * beta *...
( Tjalpha .* sign(Tj) * w ...
+ Tkalpha .* sign(Tk) * w ...
- TjpTkalpha .* sign(TjpTk) * w);

sig = (exp(A) .* cos(B) + exp(D).*cos(E) - 2)./...
( 2 * N * gam^(2*alpha) * TjxTk.^alpha);

end

function sig = charCov2(t ,N, alpha , beta, gam)
% Compute covariance matrix of
z = Arctan(imag(phi(t))/real(phi(t))),
% where phi(t) is ecf of alpha-stable random variables.
% Based on Theorem in [2].
K = length(t);
w = tan(alpha*pi/2);
calpha = gam^alpha;

Tj = repmat( t(:) , 1 , K);
Tk = repmat( t(:)' , K , 1);
Tjalpha = abs(Tj).^alpha;
Tkalpha = abs(Tk).^alpha;
TjpTk = Tj + Tk ;
TjpTkalpha = abs(TjpTk).^alpha;
TjmTk = Tj - Tk ;
TjmTkalpha = abs(TjmTk).^alpha;

B = calpha * beta *...
(-Tjalpha .* sign(Tj) * w ...
+ Tkalpha .* sign(Tk) * w ...
+ TjmTkalpha .* sign(TjmTk) * w) ;
E = calpha * beta *...
( Tjalpha .* sign(Tj) * w ...
+ Tkalpha .* sign(Tk) * w ...
- TjpTkalpha .* sign(TjpTk) * w);
F = calpha * (Tjalpha + Tkalpha);
G = -calpha * TjmTkalpha;
H = -calpha * TjpTkalpha;

```

```
sig = exp(F) .* (exp(G) .* cos(B) - exp(H) .* cos(E)) / (2*N);

end
```

C.5 Stable Distribution CDF inversion

```
function x = stblinv(u,alpha,beta,gam,delta,varargin)
%X = STBLINV(U,ALPHA,BETA,GAM,DELTA) returns values
of the inverse CDF of
% the alpha-stable distribution with characteristic
exponent ALPHA, skewness
% BETA, scale GAM, and location DELTA at the values
in the array U.
%
% This algorithm uses a combination of Newton's
method and the bisection
% method to compute the inverse cdf to a
tolerance of 1e-6;
%
% X = STBLINV(U,ALPHA,BETA,GAM,DELTA,'quick') returns
a linear interpolated
% approximation of the inverse CDF based on a
table of pre-calculated
% values. The table contains exact values at
% ALPHA = [.1 : .1: 2]
% BETA = [0: .2 : 1]
% U = [ .1 : .1 : .9 ]
% If U < .1 or U > .9, the 'quick' option approximates
the CDF with its
% asymptotic form which is given in [1], page 16,
Property 1.2.15. Results
% for U outside the interval [.1:.9] may vary.
%
% If abs(ALPHA - 1) < 1e-5, ALPHA is rounded to 1.
%
% See also: STBLRND, STBLPDF, STBLCDF, STBLFIT
%
% Reference:
% [1] G. Samorodnitsky & M. S. Taqqu (1994)
% "Stable Non-Gaussian Random Processes,
Stochastic Models with
% Infinite Variance" Chapman & Hall
%
if nargin < 5
error('stblcdf:TooFewInputs',
'Requires at least five input arguments.');
```

```

end

% Check parameters
if alpha <= 0 || alpha > 2 || ~isscalar(alpha)
error('stblcdf:BadInputs',' "alpha" must be a
scalar which lies in the interval (0,2]');
end
if abs(beta) > 1 || ~isscalar(beta)
error('stblcdf:BadInputs',' "beta" must be a
scalar which lies in the interval [-1,1]');
end
if gam < 0 || ~isscalar(gam)
error('stblcdf:BadInputs',' "gam"
must be a non-negative scalar');
end
if ~isscalar(delta)
error('stblcdf:BadInputs','
"delta" must be a scalar');
end

if (1e-5 < abs(alpha - 1) && abs(alpha - 1) < .02)
|| alpha < .02
warning('stblcdf:ScaryAlpha',...
'Difficult to approximate cdf for alpha
close to 0 or 1')
end

quick = false;
if ~isempty(varargin)
if strcmp(varargin{1},'quick')
quick = true;
end
end

% For Newton's Method
itermax = 30;
maxbisecs = 30;
tol = 1e-8;

% Return NaN for out of range parameters or probabilities.
u(u < 0 | 1 < u) = NaN;

% Check to see if you are in a simple case, if so be quick.
if alpha == 2 % Gaussian distribution
x = - 2 .* erfcinv(2*u);
x = x*gam + delta;
elseif alpha==1 && beta == 0 % Cauchy distribution
x = tan(pi*(u - .5) );
x = x*gam + delta;
elseif alpha == .5 && abs(beta) == 1 % Levy distribution

```

```

x = .5 * beta ./ (erfcinv(u)).^2;
x = x*gam + delta;
else % Gen. Case

% Flip sign of beta if necessary
if beta < 0
signBeta = -1;
u = 1-u;
beta = -beta;
else
signBeta = 1;
end

% Calculate additional shift for (M) parameterization
if abs(alpha - 1) > 1e-5
deltaM = -beta * tan(alpha*pi/2); %
else
deltaM = 0;
end

x = intGuess(alpha,beta,u);
if ~quick
% Newton's Method
F = stblcdf(x,alpha,beta,1,deltaM) - u;
diff = max(abs(F),0); % max incase of NaNs
bad = diff > tol;
iter = 1;
Fold = F;
xold = x;
while any(bad(:)) && iter < itermax

% Perform Newton step
% If Fprime = 0, step closer to origin instead
Fprime = stblpdf(x(bad),alpha,beta,1,deltaM,1e-8);
x(bad) = x(bad) - F(bad) ./ Fprime;
blowup = isinf(x) |.isnan(x);
if ~isempty(blowup)
x(blowup) = xold(blowup) / 2;
end

F(bad) = stblcdf(x(bad),alpha,beta,1,deltaM) - u(bad);

% Make sure we are getting closer, if not, do bisections until
% we do.
nocvg = abs(F) > 1.1*abs(Fold);
bisecs = 0;
while any(nocvg(:)) && (bisecs < maxbisecs)
x(nocvg) = .5*(x(nocvg) + xold(nocvg));
F(nocvg) = stblcdf(x(nocvg),alpha,beta,1,deltaM) - u(nocvg);
nocvg = abs(F) > 1.1*abs(Fold);

```



```

bisecs = bisecs + 1;
end

% Test for convergence
diff = max(abs(F),0); % max incase of NaNs
bad = diff > tol;

% Save for next iteration
xold = x;
Fold = F;
iter = iter + 1;
end
end

% Un-standardize
if abs(1 - alpha) > 1e-5
x = signBeta*(x - deltaM)*gam + delta;
else
x = signBeta*(x*gam + (2/pi) * beta * gam * log(gam)) + delta;
end

end

end

%=====
%=====
%=====

function X0 = intGuess(alpha,beta,u)
% Look-up table of percentiles of standard stable distributions
% If .1 < u < .9, Interpolates tabulated values
to obtain initial guess
% If u < .1 or u > .9 uses asymptotic formulas to make
a starting guess

utemp = u(:);
X0 = zeros(size(utemp));
alpha = max(alpha,.1);
if beta == 1
utemp(utemp < .1) = .1; % bring these into table range
end % since asyp. formulas don't
apply if beta=1.

high = (utemp > .9);
low = (utemp < .1);
middle = ~high & ~low;

```

```

% Use asymptotic formulas to guess high and low
if any(high | low)
if alpha ~= 1
C = (1-alpha) / ( gamma(2 - alpha) * cos(pi*alpha/2) );
else
C = 2/pi;
end
X0(high) = ( (1-u(high))/(C * .5 * (1 + beta)) ).^(-1/alpha);
X0(low) = -(u(low)/(C * .5 * (1 - beta))).^(-1/alpha);
end

% Use pre-calculated lookup table
if any(middle)
[Alp Bet P] = meshgrid(.1:.1:2 , 0:.2:1 , .1:.1:.9 );
stblfrac = zeros(6,20,9);
stblfrac(:,1:5,1) = ... %
[-1.890857122067030e+006    -1.074884919696010e+003
-9.039223076384694e+001    -2.645987890965098e+001
-1.274134564492298e+001;...
-1.476366405440763e+005    -2.961237538429159e+002
-3.771873580263473e+001    -1.357404219788403e+001
-7.411052003232824e+000;...
-4.686998894118387e+003    -5.145071882481552e+001
-1.151718246460839e+001    -5.524535336243413e+000
-3.611648531595958e+000;...
-2.104710824345458e+001    -3.379418096823576e+000
-1.919928049616870e+000    -1.508399002681057e+000
-1.348510542803496e+000;...
-1.267075422596289e-001    -2.597188113311268e-001
-4.004811495862077e-001    -5.385024279816432e-001
-6.64291652077534e-001;...
-1.582153175255304e-001    -3.110425775503970e-001
-4.383733961816599e-001    -5.421475800719634e-001
-6.303884905318050e-001];
stblfrac(:,6:10,1) = ...
[-7.864009406553024e+000    -5.591791397752695e+000
-4.343949435866958e+000    -3.580521076832391e+000
-3.077683537175253e+000;...
-4.988799898398770e+000    -3.787942909197120e+000
-3.103035515608863e+000    -2.675942594722292e+000
-2.394177022026705e+000;...
-2.762379160216148e+000    -2.313577186902494e+000
-2.052416861482463e+000    -1.893403771865641e+000
-1.796585983161395e+000;...
-1.284465355994317e+000    -1.267907903071982e+000
-1.279742001004255e+000    -1.309886183701422e+000
-1.349392554642457e+000;...
-7.754208907962602e-001    -8.732998811318613e-001
-9.604322013853581e-001    -1.039287445657237e+000

```

```

-1.111986321525904e+000;...
-7.089178961038225e-001    -7.814055112235459e-001
-8.502117698317242e-001    -9.169548634355569e-001
    -9.828374636178471e-001];
stblfrac(:,11:15,1) = ...
[-2.729262880847457e+000    -2.479627528870857e+000
-2.297138304998905e+000    -2.162196365947914e+000
    -2.061462692277420e+000;...
-2.202290611202918e+000    -2.070075681428623e+000
-1.979193969170630e+000    -1.917168989568703e+000
    -1.875099179801364e+000;...
-1.740583121589162e+000    -1.711775396141753e+000
-1.700465158047576e+000    -1.700212465596452e+000
    -1.707238269631509e+000;...
-1.391753942957071e+000    -1.434304119387730e+000
-1.476453646904256e+000    -1.518446568503842e+000
    -1.560864595722380e+000;...
-1.180285915835185e+000    -1.245653509438976e+000
-1.309356535558631e+000    -1.372547245869795e+000
    -1.436342854982504e+000;...
-1.048835660976022e+000    -1.115815771583362e+000
-1.184614345408666e+000    -1.256100352867799e+000
    -1.331235978799527e+000];
stblfrac(:,16:20,1) = ...
[-1.985261982958637e+000    -1.926542865732525e+000
-1.880296841910385e+000    -1.843044812063057e+000
    -1.812387604873646e+000;...
-1.846852935880107e+000    -1.828439745755405e+000
-1.817388844989596e+000    -1.812268962543248e+000
    -1.812387604873646e+000;...
-1.719534615317151e+000    -1.736176665562027e+000
-1.756931455967477e+000    -1.782079727531726e+000
    -1.812387604873646e+000;...
-1.604464355709833e+000    -1.650152416312346e+000
-1.699029550621646e+000    -1.752489822658308e+000
    -1.812387604873646e+000;...
-1.501904088536648e+000    -1.570525854475943e+000
-1.643747672313277e+000    -1.723509779436442e+000
    -1.812387604873646e+000;...
-1.411143947581252e+000    -1.497190629447853e+000
-1.591104422133556e+000    -1.695147748117837e+000
    -1.812387604873646e+000];

stblfrac(:,1:5,2) = ...
[-4.738866777987500e+002    -1.684460387562537e+001
    -5.619926961081743e+000    -3.281734135829228e+000
        -2.397479160864619e+000;...
-2.185953347160669e+001    -3.543320127025984e+000
    -1.977029667649595e+000    -1.507632281031653e+000
        -1.303310228044346e+000;...

```

```

-2.681009914911080e-001    -4.350930213152404e-001
-5.305212880041126e-001    -6.015232065896753e-001
    -6.620641788021128e-001;...
-9.503065419472154e-002    -1.947070824738389e-001
-2.987136341021804e-001    -3.973064532664002e-001
    -4.838698271554803e-001;...
-1.264483719244014e-001    -2.437377726529247e-001
-3.333750988387906e-001    -4.016893641684894e-001
    -4.577316520822721e-001;...
-1.526287733702501e-001    -2.498255243669921e-001
-3.063859169446500e-001    -3.504924054764082e-001
    -3.911254396222550e-001];
stblfrac(:,6:10,2) = ...
[-1.959508008521143e+000    -1.708174380583835e+000
-1.550822278332538e+000    -1.447013328833974e+000
    -1.376381920471173e+000;...
-1.199548019673933e+000    -1.144166826374866e+000
-1.115692821970145e+000    -1.103448361903579e+000
    -1.101126400280696e+000;...
-7.174026993828067e-001    -7.694003004766365e-001
-8.178267862332173e-001    -8.615585464741182e-001
    -9.003104216523169e-001;...
-5.579448431371428e-001    -6.215822273361273e-001
-6.771753949313707e-001    -7.267793058476849e-001
    -7.720164852674839e-001;...
-5.069548741156986e-001    -5.523620701546919e-001
-5.956554729327528e-001    -6.378655338388568e-001
    -6.796745661620428e-001;...
-4.309657384679277e-001    -4.709130419301468e-001
-5.113624096299824e-001    -5.525816075847192e-001
    -5.948321009341774e-001];
stblfrac(:,11:15,2) = ...
[-1.327391983207241e+000    -1.292811209009340e+000
-1.267812588403031e+000    -1.249132310044230e+000
    -1.234616432819130e+000;...
-1.104531584444055e+000    -1.110930462397609e+000
-1.118760810700929e+000    -1.127268239360369e+000
    -1.136171639806347e+000;...
-9.347554970493899e-001    -9.658656088352816e-001
-9.945788535033495e-001    -1.021718797792234e+000
    -1.048005562158225e+000;...
-8.141486817740096e-001    -8.541760575495752e-001
-8.929234555236560e-001    -9.311104141820112e-001
    -9.694099704722252e-001;...
-7.215886443544494e-001    -7.640354693071291e-001
-8.074261467088205e-001    -8.522003643607233e-001
    -8.988670244927735e-001;...
-6.384119892912432e-001    -6.836776839822375e-001
-7.310612144698296e-001    -7.810921001396979e-001
    -8.344269070778757e-001];

```

```

stblfrac(:,16:20,2) = ...
[-1.222879780072203e+000    -1.213041554808853e+000
 -1.204541064608597e+000    -1.197016952370690e+000
 -1.190232162899989e+000;...
-1.145449097190615e+000    -1.155224344271089e+000
-1.165719407748303e+000    -1.177246763148178e+000
 -1.190232162899989e+000;...
-1.074094694885961e+000    -1.100624477495892e+000
-1.128270402039747e+000    -1.157812818875688e+000
 -1.190232162899989e+000;...
-1.008502023575024e+000    -1.049129636922346e+000
-1.092166845038550e+000    -1.138712425453996e+000
 -1.190232162899989e+000;...
-9.480479125009214e-001    -1.000533792677121e+000
-1.057363229272293e+000    -1.119941850176443e+000
 -1.190232162899989e+000;...
-8.918931068397437e-001    -9.545526172382969e-001
-1.023797332562095e+000    -1.101496412960141e+000
 -1.190232162899989e+000];

stblfrac(:,1:5,3) = ...
[-1.354883142615948e+000    -8.855778500552980e-001
 -7.773858277863260e-001    -7.357727812399337e-001
 -7.181850957003714e-001;...
-5.193811327974376e-002    -1.633949875159595e-001
-2.617724006156590e-001    -3.392619822712012e-001
 -4.018554923458003e-001;...
-6.335376612981386e-002    -1.297738965263227e-001
-1.985319371835911e-001    -2.624863717000360e-001
 -3.174865471926985e-001;...
-9.460338726038994e-002    -1.756165596280472e-001
-2.282691311262980e-001    -2.638458905915733e-001
 -2.918110046315503e-001;...
-1.158003423724520e-001    -1.620942232133271e-001
-1.790483132028017e-001    -1.937097725890709e-001
 -2.109729530977958e-001;...
-5.695213481951577e-002    -2.485009114767256e-002
-2.455774348005581e-002    -4.243720620421176e-002
 -6.906960852184874e-002];

stblfrac(:,6:10,3) = ...
[ -7.120493514301658e-001    -7.121454153857569e-001
 -7.157018373526386e-001    -7.209253714350538e-001
 -7.265425280053609e-001;...
-4.539746445467862e-001    -4.979328472153985e-001
-5.348184073267474e-001    -5.654705188376931e-001
 -5.909430146259388e-001;...
-3.637544360366539e-001    -4.030045272659678e-001
-4.369896090801292e-001    -4.671253359013797e-001
 -4.944847533335236e-001;...
-3.167744873288179e-001    -3.408290016876749e-001

```

```

-3.649204420006245e-001    -3.894754728525021e-001
-4.146904022890949e-001;...
-2.311198638992638e-001    -2.537077422985343e-001
-2.783252370301364e-001    -3.047045003309861e-001
-3.327092628454751e-001;...
-1.000745485866474e-001    -1.334091111747126e-001
-1.681287272131953e-001    -2.038409527302062e-001
-2.404547731975402e-001];
stblfrac(:,11:15,3) = ...
[-7.317075569303094e-001    -7.359762286696208e-001
-7.392122467978279e-001    -7.414607677550720e-001
-7.428480570989012e-001;...
-6.123665499489599e-001    -6.307488506465194e-001
-6.469130897780404e-001    -6.615145568123281e-001
-6.750798357120451e-001;...
-5.198770070249209e-001    -5.439265161390062e-001
-5.671356857543234e-001    -5.899325077218274e-001
-6.127077038151078e-001;...
-4.406707089221509e-001    -4.675033009839270e-001
-4.952960990683358e-001    -5.242037261193876e-001
-5.544463409264927e-001;...
-3.623063449447594e-001    -3.935470145089454e-001
-4.265595391976379e-001    -4.615525703717921e-001
-4.988293297210071e-001;...
-2.780623638274261e-001    -3.168837529800063e-001
-3.572466721186688e-001    -3.995862986780706e-001
-4.444626893956575e-001];
stblfrac(:,16:20,3) = ...
[-7.435216571211187e-001    -7.436225251216279e-001
-7.432733099840527e-001    -7.425762029730668e-001
-7.416143171871161e-001;...
-6.880470899358724e-001    -7.008026232247697e-001
-7.137148222421971e-001    -7.271697520465581e-001
-7.416143171871161e-001;...
-6.358474023877762e-001    -6.597648782206755e-001
-6.849381555866478e-001    -7.119602076523737e-001
-7.416143171871161e-001;...
-5.863313160876512e-001    -6.202819599064874e-001
-6.568811178840162e-001    -6.969403639254603e-001
-7.416143171871159e-001;...
-5.388134824040952e-001    -5.820906647738434e-001
-6.294732446564461e-001    -6.821024214831549e-001
-7.416143171871159e-001;...
-4.925935308416445e-001    -5.449092276644302e-001
-6.026377433551201e-001    -6.674379829825384e-001
-7.416143171871159e-001];

stblfrac(:,1:5,4) = ...
[-4.719005698760254e-003    -5.039419714218448e-002
-1.108600074872916e-001    -1.646393852283324e-001

```

```

-2.088895889525075e-001; ...
-3.167687806490741e-002    -6.488347295237770e-002
-9.913854730442322e-002    -1.306663969875579e-001
-1.574578108363950e-001; ...
-6.256908981229170e-002    -1.058190431028687e-001
-1.215669874255146e-001    -1.261149689648148e-001
-1.284283108027729e-001; ...
-7.132464704948761e-002    -5.885471032381771e-002
-3.846810486653290e-002    -2.801768649688129e-002
-2.615407079824540e-002; ...
1.186775035989228e-001    1.847231744541209e-001
1.899666578065291e-001    1.756596652192159e-001
1.538218851318199e-001; ...
1.359937191266603e+000    7.928324704017256e-001
6.068350758065271e-001    4.949176895753282e-001
4.117787224185477e-001];
stblfrac(:,6:10,4) = ...
[-2.445873831127209e-001    -2.729819770922066e-001
-2.951510874462016e-001    -3.121233685073350e-001
-3.249196962329062e-001; ...
-1.797875581290475e-001    -1.986122400020671e-001
-2.148458045681510e-001    -2.292024720743768e-001
-2.422125650878785e-001; ...
-1.318108373643454e-001    -1.372885008966837e-001
-1.450218673440198e-001    -1.548461140242879e-001
-1.664940537646226e-001; ...
-3.037902421859952e-002    -3.894619676380785e-002
-5.076849313651704e-002    -6.518223105549245e-002
-8.178056142331483e-002; ...
1.287679439328719e-001    1.022243387982872e-001
7.488543991005173e-002    4.698265181928261e-002
1.852002327642577e-002; ...
3.435869264264112e-001    2.844376471729288e-001
2.312306852681522e-001    1.820841981890349e-001
1.357181057787019e-001];
stblfrac(:,11:15,4) = ...
[-3.344714240325961e-001    -3.415532212363377e-001
-3.467713617249639e-001    -3.505859000173167e-001
-3.533413466958321e-001; ...
-2.542699931601989e-001    -2.656748454748664e-001
-2.766656461455947e-001    -2.874428940341864e-001
-2.981872822548070e-001; ...
-1.796994139325742e-001    -1.942454974557965e-001
-2.099854734361004e-001    -2.268483937252861e-001
-2.448403779828917e-001; ...
-1.003134231215546e-001    -1.206343411798188e-001
-1.426762955132322e-001    -1.664453845103147e-001
-1.920257997377931e-001; ...
-1.062008675791458e-002    -4.062891141128176e-002
-7.175196683590498e-002    -1.042870733773311e-001

```

```

-1.385948877988075e-001;...
9.117291945474759e-002    4.766184332000264e-002
4.481886485253039e-003    -3.904933750228177e-002
-8.364689014849616e-002];
stblfrac(:,16:20,4) = ...
[-3.552947623689004e-001    -3.566384591258251e-001
-3.575167387322836e-001    -3.580387843935552e-001
-3.582869092425832e-001;...
-3.090746307371333e-001    -3.202900038682522e-001
-3.320450798333745e-001    -3.445973947956370e-001
-3.582869092425832e-001;...
-2.640470286750166e-001    -2.846415660837839e-001
-3.069024734642628e-001    -3.312464672828315e-001
-3.582869092425832e-001;...
-2.195942670864279e-001    -2.494428999135824e-001
-2.820166786810741e-001    -3.179740384308457e-001
-3.582869092425832e-001;...
-1.751227987938045e-001    -2.144432379167035e-001
-2.573138196343415e-001    -3.047716553689650e-001
-3.582869092425832e-001;...
-1.301133939768983e-001    -1.794049920724848e-001
-2.327202766583559e-001    -2.916310469293936e-001
-3.582869092425832e-001];

stblfrac(:,1:5,5) = ...
[
0
0
0;...
-2.998229841415443e-002    -3.235136568035350e-002
-1.058934315424071e-002    1.472786013654386e-002
3.649529125352272e-002;...
-4.911181618214269e-004    7.928758678692660e-002
1.295711243349632e-001    1.575625247967377e-001
1.726794061650541e-001;...
6.444732609572413e-001    5.412205715497974e-001
4.864603927210872e-001    4.457073928551408e-001
4.118964225372133e-001;...
4.884639795042095e+000    1.686842470765597e+000
1.132342494635284e+000    8.944978064032267e-001
7.538011200000044e-001;...
2.410567057697245e+001    4.005534670805399e+000
2.144263118197206e+000    1.518214626927320e+000
1.198109338317733e+000];
stblfrac(:,6:10,5) = ...
[
0
0
0;...
5.320761222262883e-002    6.497369053185199e-002
7.235439352353751e-002    7.603800885095309e-002
7.671459793802817e-002;...

```



```

1.799982238321182e-001    1.821699713013862e-001
1.806145618464317e-001    1.761248753943454e-001
1.691770293512301e-001;...
3.823074983529713e-001    3.554905959697276e-001
3.305043126978712e-001    3.066571802106021e-001
2.834017043112906e-001;...
6.558265419066330e-001    5.806408912949470e-001
5.191065509143589e-001    4.663489244354866e-001
4.194539705064985e-001;...
9.966378800612080e-001    8.532685386168033e-001
7.427048697651345e-001    6.524693172360032e-001
5.756299950589361e-001];
stblfrac(:,11:15,5) = ...
[
    0
    0
    0
    0;...

7.500001602159387e-002    7.139599669434762e-002
6.628276247821394e-002    5.992932695316782e-002
5.250925428603021e-002;...
1.600901411017374e-001    1.491003610537801e-001
1.363865273697878e-001    1.220722641614886e-001
1.062191001109524e-001;...
2.602853501366307e-001    2.369238065872132e-001
2.129824521942899e-001    1.881563959610275e-001
1.621474808586950e-001;...
3.765099860312678e-001    3.361566147323812e-001
2.973499640484341e-001    2.592283952427927e-001
2.210255604589869e-001;...
5.079606300067100e-001    4.466711396792393e-001
3.897746494263863e-001    3.357416130711989e-001
2.832892169418335e-001];
stblfrac(:,16:20,5) = ...
[
    0
    0
    0
    0;...

4.411421669339249e-002    3.476266163507976e-002
2.439917920106283e-002    1.289010976694223e-002
0;...

8.881586460416716e-002    6.976629777350905e-002
4.886974404989612e-002    2.578932638717129e-002
0;...

1.346349888095220e-001    1.052403813710735e-001
7.348119932151805e-002    3.870673240105876e-002
0;...

1.820030836908522e-001    1.413881485626739e-001
9.829989964989198e-002    5.165115573609639e-002
0;...

2.312355801087936e-001    1.783807793433976e-001
1.233869208812706e-001    6.463145748462040e-002
9.714451465470120e-017];

```

```

stblfrac(:,1:5,6) = ...
[ 4.719005698760275e-003    5.039419714218456e-002
  1.108600074872919e-001    1.646393852283322e-001
    2.088895889525074e-001;...
1.944613194060750e-001    3.117984496788369e-001
  3.615078716560812e-001    3.879646155737581e-001
    4.042606354602197e-001;...
3.045958300133999e+000    1.315675725057089e+000
  9.757973307352019e-001    8.294361410388060e-001
    7.456405896421690e-001;...
2.339312510820383e+001    3.858569195402605e+000
  2.091507439545032e+000    1.515362821077606e+000
    1.231804842218289e+000;...
1.231812404655975e+002    9.151933726881032e+000
  3.856468345925451e+000    2.470027172456050e+000
    1.862167039303084e+000;...
5.049829135345403e+002    1.890722475322573e+001
  6.427275565975617e+000    3.715903402980179e+000
    2.636417882085815e+000];
stblfrac(:,6:10,6) = ...
[ 2.445873831127209e-001    2.729819770922065e-001
  2.951510874462016e-001    3.121233685073347e-001
    3.249196962329060e-001;...
4.152379986226543e-001    4.229018705591941e-001
  4.280900470005300e-001    4.311273812611276e-001
    4.321442286112657e-001;...
6.900226415397631e-001    6.495436520935480e-001
  6.180526887451320e-001    5.921654464012007e-001
    5.697923159645174e-001;...
1.060749495885882e+000    9.442937075476816e-001
  8.583603822642385e-001    7.911221543980916e-001
    7.360251815557063e-001;...
1.521067254392224e+000    1.300039377551776e+000
  1.142711537858461e+000    1.023045102736937e+000
    9.273664178094935e-001;...
2.065989355542487e+000    1.711228437455139e+000
  1.466088158475343e+000    1.283765226486882e+000
    1.140575450959062e+000];
stblfrac(:,11:15,6) = ...
[3.344714240325963e-001    3.415532212363379e-001
  3.467713617249641e-001    3.505859000173170e-001
    3.533413466958320e-001;...
4.312423594533669e-001    4.285591238013830e-001
  4.242644840754073e-001    4.185310514289916e-001
    4.115050794489342e-001;...
5.495326577846258e-001    5.304020801294532e-001
  5.116943409858906e-001    4.928954730588648e-001
    4.736165965702772e-001;...

```

```

6.890778676134198e-001    6.476526200515113e-001
6.099033923678876e-001    5.744600864566568e-001
5.402514096915735e-001;...
8.477633920324498e-001    7.792812067953944e-001
7.185943530039393e-001    6.633207377171386e-001
6.116407715135426e-001;...
1.023262411940948e+000    9.237922892835746e-001
8.369566524681974e-001    7.591595457820644e-001
6.877508180861301e-001];
stblfrac(:,16:20,6) = ...
[ 3.552947623689000e-001    3.566384591258254e-001
 3.575167387322835e-001    3.580387843935554e-001
 3.582869092425831e-001;...
4.032875933324668e-001    3.939222836649399e-001
3.833860261287606e-001    3.715758694363207e-001
3.582869092425831e-001;...
4.535361612745278e-001    4.323485980953122e-001
4.097162006469898e-001    3.852184728042033e-001
3.582869092425835e-001;...
5.063904595668142e-001    4.720865286037160e-001
4.365637761840112e-001    3.989743423180101e-001
3.582869092425835e-001;...
5.620594176198462e-001    5.132627179036522e-001
4.639774715385669e-001    4.128508865888630e-001
3.582869092425835e-001;...
6.206265009880273e-001    5.559603894356728e-001
4.919976875425384e-001    4.268552022160075e-001
3.582869092425835e-001];

stblfrac(:,1:5,7) = ...
[ 1.354883142615939e+000    8.855778500552969e-001
 7.773858277863266e-001    7.357727812399328e-001
 7.181850957003700e-001;...
2.264297017396562e+001    3.703766301758638e+000
2.034998948698223e+000    1.510923485095245e+000
1.265729978744353e+000;...
1.955956459466261e+002    1.118917023817671e+001
4.357570503031440e+000    2.718083521990130e+000
2.041945502327640e+000;...
1.131527106972301e+003    2.742019413138009e+001
8.094356141096943e+000    4.405625422851678e+000
3.045873292912599e+000;...
4.991370610374878e+003    5.832596523112534e+001
1.361736440227531e+001    6.617793943005997e+000
4.277065691957527e+000;...
1.808482789458792e+004    1.120299053944505e+002
2.131886896428897e+001    9.395528700779570e+000
5.735282952993835e+000];
stblfrac(:,6:10,7) = ...
[ 7.120493514301658e-001    7.121454153857567e-001

```

```

7.157018373526382e-001    7.209253714350531e-001
7.265425280053608e-001; ...
1.126910935459891e+000    1.039315711942880e+000
9.801156996469297e-001    9.380990288559633e-001
9.070002633955093e-001; ...
1.682687096145072e+000    1.462088170281394e+000
1.313508264506275e+000    1.206803763884095e+000
1.126395471042167e+000; ...
2.368493556832589e+000    1.968378518204384e+000
1.704951233806636e+000    1.518043793772535e+000
1.377948007790416e+000; ...
3.176211386678905e+000    2.549432728119129e+000
2.146593646702069e+000    1.865193645178458e+000
1.656315874739094e+000; ...
4.099439855675913e+000    3.198582996879541e+000
2.632582798272859e+000    2.243339709179312e+000
1.957469852365064e+000];
stblfrac(:,11:15,7) = ...
[ 7.317075569303093e-001    7.359762286696208e-001
7.392122467978273e-001    7.414607677550722e-001
7.428480570989009e-001; ...
8.829463516299942e-001    8.633779161543368e-001
8.465599716104961e-001    8.313215935120923e-001
8.168794983145117e-001; ...
1.063360967480519e+000    1.012144436660489e+000
9.690437805764626e-001    9.314651792280744e-001
8.975270882378618e-001; ...
1.268363069256580e+000    1.179563109954373e+000
1.105319244270462e+000    1.041384485194864e+000
9.846979577532636e-001; ...
1.493891969504980e+000    1.362797559741365e+000
1.253624580847262e+000    1.160149469096889e+000
1.078008118654219e+000; ...
1.736744887299007e+000    1.559416515511960e+000
1.412280239489399e+000    1.286729855523644e+000
1.176933895080190e+000];
stblfrac(:,16:20,7) = ...
[ 7.435216571211178e-001    7.436225251216276e-001
7.432733099840527e-001    7.425762029730666e-001
7.416143171871158e-001; ...
8.027015701907034e-001    7.884022863227798e-001
7.736657968963813e-001    7.581862145381915e-001
7.416143171871158e-001; ...
8.658237613571567e-001    8.352619776464638e-001
8.049334692839693e-001    7.740056420537431e-001
7.416143171871158e-001; ...
9.329399521299938e-001    8.842632875709708e-001
8.371061471443788e-001    7.900396709438159e-001
7.416143171871157e-001; ...
1.003953952010710e+000    9.354146255148074e-001

```

```

8.702022492276336e-001    8.062927602676150e-001
7.416143171871157e-001; ...
1.078670034479511e+000    9.886802003678273e-001
9.042295460529033e-001    8.227686378257326e-001
7.416143171871157e-001];

stblfrac(:,1:5,8) = ...
[4.738866777987514e+002    1.684460387562540e+001
5.619926961081758e+000    3.281734135829232e+000
2.397479160864624e+000; ...
4.841681688643794e+003    5.491635522391771e+001
1.256979234254407e+001    6.069209132601843e+000
3.940274296039883e+000; ...
3.154616792561625e+004    1.420805372229245e+002
2.403953052063284e+001    9.998426062380954e+000
5.930362539243756e+000; ...
1.520631636586534e+005    3.148956061770992e+002
4.132943146104890e+001    1.518515134801384e+001
8.367182529059960e+000; ...
5.901656732159231e+005    6.246491282963873e+002
6.581680474603525e+001    2.173557079848703e+001
1.125045444319795e+001; ...
1.944624278667431e+006    1.139848804168331e+003
9.894809619823921e+001    2.974824391888133e+001
1.458002371721213e+001];

stblfrac(:,6:10,8) = ...
[1.959508008521145e+000    1.708174380583837e+000
1.550822278332539e+000    1.447013328833976e+000
1.376381920471174e+000; ...
2.963447020215305e+000    2.423693540860402e+000
2.089182215079736e+000    1.865572849084425e+000
1.708118159360888e+000; ...
4.190132768594454e+000    3.268280841745006e+000
2.710662024401290e+000    2.341995909523891e+000
2.082469140437107e+000; ...
5.624308785058203e+000    4.226708866462347e+000
3.402197103627229e+000    2.865360079281767e+000
2.490393899977397e+000; ...
7.254212029229660e+000    5.287806421003054e+000
4.154585933912857e+000    3.428194997160839e+000
2.925780747207696e+000; ...
9.070365685373144e+000    6.442950257298201e+000
4.960971490178073e+000    4.025088868546689e+000
3.384287797654701e+000];

stblfrac(:,11:15,8) = ...
[ 1.327391983207241e+000    1.292811209009341e+000
1.267812588403031e+000    1.249132310044230e+000
1.234616432819130e+000; ...
1.593041126030172e+000    1.506471132927683e+000
1.439628954887186e+000    1.386580264484466e+000

```

```

1.343153406231364e+000;...
1.891158929781140e+000    1.745070641877115e+000
  1.630251730907927e+000    1.537630629971792e+000
  1.460938380853296e+000;...
2.214464603850502e+000    2.003098342270666e+000
  1.835905829230373e+000    1.700021765831942e+000
  1.586823477367793e+000;...
2.557944985263177e+000    2.276562749626175e+000
  2.053593165082403e+000    1.871725504345519e+000
  1.719630879614922e+000;...
2.918103805585008e+000    2.562588803694463e+000
2.281050180010934e+000    2.051085944176459e+000
1.858294826115218e+000];
stblfrac(:,16:20,8) = ...
[ 1.222879780072204e+000    1.213041554808854e+000
  1.204541064608597e+000    1.197016952370690e+000
  1.190232162899990e+000;...
1.306371038922589e+000    1.274091491606534e+000
1.244744203398707e+000    1.217124809801410e+000
1.190232162899990e+000;...
1.395630981221581e+000    1.338301797693731e+000
  1.286320343916442e+000    1.237570697847646e+000
  1.190232162899990e+000;...
1.490188322141933e+000    1.405530485165501e+000
1.329245194088195e+000    1.258353899045780e+000
  1.190232162899990e+000;...
1.589489775546923e+000    1.475587597649461e+000
  1.373481210080780e+000    1.279472666002594e+000
  1.190232162899990e+000;...
1.692973560150181e+000    1.548256386823049e+000
1.418980226656540e+000    1.300924242481222e+000
1.190232162899990e+000];

stblfrac(:,1:5,9) = ...
[1.890857122067037e+006    1.074884919696010e+003
9.039223076384690e+001    2.645987890965103e+001
1.274134564492299e+001;...
1.434546473316804e+007    2.987011338973518e+003
1.804473474220022e+002    4.487048929338575e+001
1.960113433547389e+001;...
7.716266115204613e+007    6.969521346220721e+003
3.196657990381036e+002    6.941784107578008e+001
  2.798990029407097e+001;...
3.253192550565641e+008    1.437315176424486e+004
  5.205876769957880e+002    1.006582035946658e+002
  3.790739646062081e+001;...
1.143638705833100e+009    2.703823367877713e+004
  7.964291266167923e+002    1.391051003571698e+002
  4.935349274736288e+001;...
3.492208269966229e+009    4.737075925045248e+004

```

```

1.161019167208514e+003    1.852377745522907e+002
6.232811767701676e+001];
stblfrac(:,6:10,9) = ...
[7.864009406553027e+000    5.591791397752693e+000
4.343949435866960e+000    3.580521076832391e+000
3.077683537175252e+000;...
1.132727408868559e+001    7.671280872680232e+000
5.732691330034323e+000    4.573075545294608e+000
3.818589092027862e+000;...
1.533578393202605e+001    9.991349773961725e+000
7.243609507849516e+000    5.634462725204553e+000
4.601857009791827e+000;...
1.985701175129152e+001    1.252691966593449e+001
8.859346059355138e+000    6.752431092162364e+000
5.418366793527828e+000;...
2.486500490402286e+001    1.525895955988075e+001
1.056731639206889e+001    7.918478700695184e+000
6.262067266019560e+000;...
3.033836510475647e+001    1.817240938152932e+001
1.235792736188858e+001    9.126360342186048e+000
7.128676006881803e+000];
stblfrac(:,11:15,9) = ...
[2.729262880847459e+000    2.479627528870858e+000
2.297138304998906e+000    2.162196365947915e+000
2.061462692277420e+000;...
3.297130126832188e+000    2.920640582387343e+000
2.640274592919582e+000    2.426998377788287e+000
2.262233765245289e+000;...
3.893417077901593e+000    3.382471282597797e+000
2.999860062957988e+000    2.705234908082859e+000
2.473610569743775e+000;...
4.510891038980249e+000    3.859051363710381e+000
3.370702720510665e+000    2.992693808833481e+000
2.692636527934335e+000;...
5.144949915764652e+000    4.346635348399592e+000
3.749599176843221e+000    3.286641675099088e+000
2.917178603817272e+000;...
5.792462636377325e+000    4.842756648977701e+000
4.134472567050430e+000    3.585273662390985e+000
3.145733197974777e+000];
stblfrac(:,16:20,9) = ...
[1.985261982958638e+000    1.926542865732524e+000
1.880296841910385e+000    1.843044812063057e+000
1.812387604873647e+000;...
2.133064562958712e+000    2.029912595114798e+000
1.945516531961286e+000    1.874392545595589e+000
1.812387604873647e+000;...
2.288441176274372e+000    2.137883347336651e+000
2.012884307837858e+000    1.906295529437326e+000
1.812387604873647e+000;...

```

```

2.449737610939970e+000    2.249772716121334e+000
2.082221357100924e+000    1.938735806854783e+000
    1.812387604873647e+000;...
2.615585030563546e+000    2.364937633815368e+000
    2.153342270485199e+000    1.971693892149562e+000
    1.812387604873647e+000;...
2.784907129216124e+000    2.482804054400846e+000
    2.226062706102394e+000    2.005149380181030e+000
    1.812387604873647e+000];

%%% Interpolate to find initial guess
[alpIn betIn uIn] = meshgrid(alpha,beta,utemp(middle));
X0(middle) = interp3(Alp,Bet,P,stblfrac,alpIn, betIn, ...
uIn, 'linear');

end

X0 = reshape(X0,size(u));

end

```


Appendix D

MATLAB Code for Applications

D.1 Speckle Filtering

```
function Ims=ImFilt(Ime,Ws)
% B-scan images filtering using HG0
% input parameters: ImageToBeProcessed and WindowSize mask
% usage: FilteredImage=sem1Asim(InputImage,WindowSize);
tic;
Wsm=Ws-1;
Ims=zeros(size(Ime));
[Nf,Nc]= size(Ime);
% floor rounds down; ceil rounds up; round nearest

Imm=ones(Ws,Ws);%Imm=128*ones(Ws,Ws);

for i=1:Nf-Wsm
for j=1:Nc-Wsm
%Avoiding gaps
%by defining init and end of window process
inih=i;finh=inih+Wsm;
iniv=j;finv=iniv+Wsm;
WinProc=Ime(inih:finh,iniv:finv);
fp = abs(FSegPar(WinProc));
mu = mean(WinProc(:));
Wm = mu*Imm;
Ims(inih:finh,iniv:finv)= uint8(Wm + fp*(double(WinProc)-Wm));
end;
end
% Double precission to integer values in matrix
Ims=uint8(Ims);
Tc=toc;
disp(Tc);
figure(1);subplot(224);image(Ims);
colormap(gray(256));
end

function SegPar= FSegPar(WinProc)
% segmenting by using skewness positive=blood
% negative either muscle or interface
```

```

Vproc=double(WinProc(:));

mu=mean(Vproc);
sg=std(Vproc);

Z=Vproc-mu;
if sg==0
SegPar=0;
else
Z=Z/sg;
m3=mean(Z.^3);
if abs(m3)< 1.09
SegPar=sign(m3)*(-0.876*abs(m3)+1.0053);
else
SegPar=0.04;
end
end
end

```

D.2 Image Segmentation

```

function Ims=segm2Asim(Ime,Ws,gr)
% B-scan images segmentation using skewness
% input parameters: ImageToBeProcessed and WindowSize mask
% usage: SegmentedImage=sem1Asim(InputImage,WindowSize);
tic;
Wsm=Ws-1;
Ims=zeros(size(Ime));
[Nf,Nc]= size(Ime);
% floor rounds down; ceil rounds up; round nearest

Imb=zeros(Ws,Ws);Imm=128*ones(Ws,Ws);

for i=1:Nf-Wsm
for j=1:Nc-Wsm
%Avoiding gaps
%by defining init and end of window process
inih=i;finh=inih+Wsm;
iniv=j;finv=iniv+Wsm;
WinProc=Ime(inih:finh,iniv:finv);
mu=mean(WinProc(:));
SegPar= FSegPar(WinProc);
if mu > 30
Ims(inih:finh,iniv:finv)=Imm+(SegPar*double(WinProc));
else
Ims(inih:finh,iniv:finv)=Imb+(SegPar*double(WinProc));
end

```

```
end;
end
% Double precission to integer values in matrix
Ims=uint8(Ims);
Tc=toc;
disp(Tc);
figure(3);subplot(221);image(Ime);subplot(2,2,gr);image(Ims);
colormap(gray(256));
end

function SegPar= FSegPar(WinProc)
% segmenting by using skewness positive=blood
% negative either muscle or interface
Vproc=double(WinProc(:));

mu=mean(Vproc);
sg=std(Vproc);
Z=Vproc-mu;
if sg==0
SegPar=0;
else
Z=Z/sg;
m3=mean(Z.^3);
if abs(m3)< 1.09
SegPar=sign(m3)*(-0.876*abs(m3)+1.0053);
else
SegPar=sign(m3)*0.04;
end
end
end
```


Bibliography

- Agrawal, Rajeev and Karmeshu (2007). "Ultrasonic backscattering in tissue: characterization through Nakagami-generalized inverse Gaussian distribution". In: *Computers in Biology and Medicine* 37.2, pp. 166–172. DOI: [10.1016/j.compbiomed.2005.11.003](https://doi.org/10.1016/j.compbiomed.2005.11.003). URL: <http://dx.doi.org/10.1016/j.compbiomed.2005.11.003>.
- Anfinson, S.N., A.P. Doulgeris, and T. Eltoft (2008). "Estimation of the Equivalent Number of Looks in Polarimetric SAR Imagery". In: *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*. Vol. 4, pp. IV –487–IV –490. DOI: [10.1109/IGARSS.2008.4779764](https://doi.org/10.1109/IGARSS.2008.4779764).
- Arsenault, H. H. and G. April (1976). "Properties of speckle integrated with a finite aperture and logarithmically transformed". In: *J. Opt. Soc. Am.* 66.11, pp. 1160–1163. DOI: [10.1364/JOSA.66.001160](https://doi.org/10.1364/JOSA.66.001160). URL: <http://www.osapublishing.org/abstract.cfm?URI=josa-66-11-1160>.
- Barakat, Richard (1986). "Weak-scatterer generalization of the K-density function with application to laser scattering in atmospheric turbulence". In: *Journal of the Optical Society of America A* 3.4, p. 401. DOI: [10.1364/josaa.3.000401](https://doi.org/10.1364/josaa.3.000401). URL: <http://dx.doi.org/10.1364/JOSAA.3.000401>.
- Berger, N. E. (1991). "Polydisperse scattering theory and comparisons with data for red blood cells". In: *The Journal of the Acoustical Society of America* 89.3, p. 1394. DOI: [10.1121/1.400540](https://doi.org/10.1121/1.400540). URL: <http://dx.doi.org/10.1121/1.400540>.
- Destrempes, Francois and Guy Cloutier (2010). "A Critical Review and Uniformized Representation of Statistical Distributions Modeling the Ultrasound Echo Envelope". In: *Ultrasound in Medicine & Biology* 36.7, pp. 1037–1051. DOI: [10.1016/j.ultrasmedbio.2010.04.001](https://doi.org/10.1016/j.ultrasmedbio.2010.04.001). URL: <http://dx.doi.org/10.1016/j.ultrasmedbio.2010.04.001>.
- Dubey, SatyaD. (1970). "Compound gamma, beta and F distributions". English. In: *Metrika* 16.1, pp. 27–31. ISSN: 0026-1335. DOI: [10.1007/BF02613934](https://doi.org/10.1007/BF02613934). URL: <http://dx.doi.org/10.1007/BF02613934>.
- Dutt, V. and J. F. Greenleaf (1994). "Ultrasound Echo Envelope Analysis Using a Homodyned K Distribution Signal Model". In: *Ultrasonic Imaging* 16.4, pp. 265–287. DOI: [10.1177/016173469401600404](https://doi.org/10.1177/016173469401600404). URL: <http://dx.doi.org/10.1177/016173469401600404>.
- Dutt, Vinayak (1995). "STATISTICAL ANALYSIS OF ULTRASOUND ECHO ENVELOPE". PhD thesis. Mayo Graduate School. URL: <http://www.mayo.edu/research/documents/duttpdf/doc-10026849>.
- (1996). "Statistics of the log-compressed echo envelope". In: *The Journal of the Acoustical Society of America* 99.6, p. 3817. DOI: [10.1121/1.414999](https://doi.org/10.1121/1.414999). URL: <http://dx.doi.org/10.1121/1.414999>.
- Eltoft, T. (2005). "The Rician inverse Gaussian distribution: a new model for non-Rayleigh signal amplitude statistics". In: *IEEE Transactions on Image Processing* 14.11, pp. 1722–1735. DOI: [10.1109/TIP.2005.857281](https://doi.org/10.1109/TIP.2005.857281). URL: <http://dx.doi.org/10.1109/TIP.2005.857281>.

- Eltoft, T. (2006a). "A new approach to modeling signal amplitude statistics by the K distributions". In: *Signal Processing Symposium, 2006. NORSIG 2006. Proceedings of the 7th Nordic*, pp. 62–65. DOI: [10.1109/NORSIG.2006.275277](https://doi.org/10.1109/NORSIG.2006.275277).
- (2006b). "Modeling the amplitude statistics of ultrasonic images". In: *Medical Imaging, IEEE Transactions on* 25.2, pp. 229–240. ISSN: 0278-0062. DOI: [10.1109/TMI.2005.862664](https://doi.org/10.1109/TMI.2005.862664).
- (2006c). "Modeling ultrasound images with the generalized K model". In: *Signal Processing Conference, 2006 14th European*, pp. 1–5.
- Frery, A. C. et al. (1999). "Models for Synthetic Aperture Radar Image Analysis". In: *Resenhas (IME-USP)* 4.1, pp. 45–77.
- Frery, A.C. et al. (1997). "A model for extremely heterogeneous clutter". In: *Geoscience and Remote Sensing, IEEE Transactions on* 35.3, pp. 648–659. ISSN: 0196-2892. DOI: [10.1109/36.581981](https://doi.org/10.1109/36.581981).
- Goodman, J. W. (1976). "Some fundamental properties of speckle*". In: *J. Opt. Soc. Am.* 66.11, pp. 1145–1150. DOI: [10.1364/JOSA.66.001145](https://doi.org/10.1364/JOSA.66.001145). URL: <http://www.osapublishing.org/abstract.cfm?URI=josa-66-11-1145>.
- Hawley, S., T. Kays, and V. Twersky (1967). "Comparison of distribution functions from scattering data on different sets of spheres". In: *IEEE Trans. Antennas Propagat.* 15.1, pp. 118–135. DOI: [10.1109/tap.1967.1138847](https://doi.org/10.1109/tap.1967.1138847). URL: <http://dx.doi.org/10.1109/TAP.1967.1138847>.
- Hoeffding, Wassily et al. (1955). "Limit Distributions for Sums of Independent Random Variables." In: *Journal of the American Statistical Association* 50.272, p. 1343. DOI: [10.2307/2281234](https://doi.org/10.2307/2281234). URL: <http://dx.doi.org/10.2307/2281234>.
- Insana, Michael F. (1986). "Analysis Of Ultrasound Image Texture Via Generalized Rician Statistics". In: *Optical Engineering* 25.6, p. 256743. DOI: [10.1117/12.7973900](https://doi.org/10.1117/12.7973900). URL: <http://dx.doi.org/10.1117/12.7973900>.
- Jakeman, E (1980). "On the statistics of K-distributed noise". In: *J. Phys. A: Math. Gen.* 13.1, pp. 31–48. DOI: [10.1088/0305-4470/13/1/006](https://doi.org/10.1088/0305-4470/13/1/006). URL: <http://dx.doi.org/10.1088/0305-4470/13/1/006>.
- Jakeman, E. and P. Pusey (1976). "A model for non-Rayleigh sea echo". In: *IEEE Trans. Antennas Propagat.* 24.6, pp. 806–814. DOI: [10.1109/tap.1976.1141451](https://doi.org/10.1109/tap.1976.1141451). URL: <http://dx.doi.org/10.1109/TAP.1976.1141451>.
- Jakeman, E. and R. J. A. Tough (1987). "Generalized K distribution: a statistical model for weak scattering". In: *Journal of the Optical Society of America A* 4.9, p. 1764. DOI: [10.1364/josaa.4.001764](https://doi.org/10.1364/josaa.4.001764). URL: <http://dx.doi.org/10.1364/JOSAA.4.001764>.
- Kullback, S. and R. A. Leibler (1951). "On Information and Sufficiency". In: *Ann. Math. Statist.* 22.1, pp. 79–86. DOI: [10.1214/aoms/1177729694](https://doi.org/10.1214/aoms/1177729694). URL: <http://dx.doi.org/10.1214/aoms/1177729694>.
- Kuruoglu, E.E. and J. Zerubia (2004). "Modeling SAR images with a generalization of the Rayleigh distribution". In: *Image Processing, IEEE Transactions on* 13.4, pp. 527–533. ISSN: 1057-7149. DOI: [10.1109/TIP.2003.818017](https://doi.org/10.1109/TIP.2003.818017).
- Lord, R. D. (1954). "The Use of the Hankel Transform in Statistics I. General Theory and Examples". In: *Biometrika* 41.1/2, p. 44. DOI: [10.2307/2333004](https://doi.org/10.2307/2333004). URL: <http://dx.doi.org/10.2307/2333004>.
- Lucas, R. J. and V. Twersky (1987). "Inversion of ultrasonic scattering data for red blood cell suspensions under different flow conditions". In: *The Journal of the Acoustical Society of America* 82.3, p. 794. DOI: [10.1121/1.395276](https://doi.org/10.1121/1.395276). URL: <http://dx.doi.org/10.1121/1.395276>.

- Mamou, Jonathan and Michael L. Oelze, eds. (2013). *Quantitative Ultrasound in Soft Tissues*. Springer Science Business Media. DOI: [10.1007/978-94-007-6952-6](https://doi.org/10.1007/978-94-007-6952-6). URL: <http://dx.doi.org/10.1007/978-94-007-6952-6>.
- Mejail, M. E. et al. (2000). "Parametric roughness estimation in amplitude SAR images under the multiplicative model". In: *Revista de Teledetección* 13, pp. 37–49.
- Mejail, M. E. et al. (2001). "Approximation of distributions for SAR Images: proposal, evaluation and practical consequences". In: *Latin American Applied Research* 31, pp. 83–92.
- Mejail, M. E. et al. (2003). "Classification of SAR Images Using a General and Tractable Multiplicative Model". In: *International Journal of Remote Sensing* 24.18, pp. 3565–3582. DOI: <http://dx.doi.org/10.1080/0143116021000053274>.
- Moser, G., J. Zerubia, and S.B. Serpico (2006). "SAR amplitude probability density function estimation based on a generalized Gaussian model". In: *Image Processing, IEEE Transactions on* 15.6, pp. 1429–1442. ISSN: 1057-7149. DOI: [10.1109/TIP.2006.871124](https://doi.org/10.1109/TIP.2006.871124).
- Mujal-Rosas, R. et al. (2007). "Electromagnetic analysis of the induction motor with spiral sheet rotor". In: *Power Electronics and Applications, 2007 European Conference on*, pp. 1–10. DOI: [10.1109/EPE.2007.4417465](https://doi.org/10.1109/EPE.2007.4417465).
- Muller, H.-J. and R. Pac (1999). "G-statistics for scaled SAR data". In: *Geoscience and Remote Sensing Symposium, 1999. IGARSS '99 Proceedings. IEEE 1999 International*. Vol. 2, 1297–1299 vol.2. DOI: [10.1109/IGARSS.1999.774610](https://doi.org/10.1109/IGARSS.1999.774610).
- Nakagami, Minoru (1960). "The m-Distribution—A General Formula of Intensity Distribution of Rapid Fading". In: *Statistical Methods in Radio Wave Propagation*. Elsevier BV, pp. 3–36. DOI: [10.1016/B978-0-08-009306-2.50005-4](https://doi.org/10.1016/B978-0-08-009306-2.50005-4). URL: <http://dx.doi.org/10.1016/B978-0-08-009306-2.50005-4>.
- Navarrete, Hugo et al. (2002). "Ultrasound image filtering using the multiplicative model". In: *Medical Imaging 2002: Ultrasonic Imaging and Signal Processing*. Ed. by Michael F. Insana and William F. Walker. SPIE-Intl Soc Optical Eng. DOI: [10.1117/12.462171](https://doi.org/10.1117/12.462171). URL: <http://dx.doi.org/10.1117/12.462171>.
- Nillesen, Maartje M. et al. (2008). "Modeling Envelope Statistics of Blood and Myocardium for Segmentation of Echocardiographic Images". In: *Ultrasound in Medicine & Biology* 34.4, pp. 674–680. DOI: [10.1016/j.ultrasmedbio.2007.10.008](https://doi.org/10.1016/j.ultrasmedbio.2007.10.008). URL: <http://dx.doi.org/10.1016/j.ultrasmedbio.2007.10.008>.
- Nolan, J. P. (2015). *Stable Distributions - Models for Heavy Tailed Data*. In progress, Chapter 1 online at academic2.american.edu/~jpnolan. Boston: Birkhauser.
- Oelze, M.L., W.D. O'Brien, and J.F. Zachary (2007). "Quantitative Ultrasound Assessment of Breast Cancer Using a Multiparameter Approach". In: *2007 IEEE Ultrasonics Symposium Proceedings*. Institute of Electrical & Electronics Engineers (IEEE). DOI: [10.1109/ultsym.2007.250](https://doi.org/10.1109/ultsym.2007.250). URL: <http://dx.doi.org/10.1109/ULTSYM.2007.250>.
- Oliver, Christopher and Shaun Quegan (2004). *Understanding Synthetic Aperture Radar Images*. SciTech Publishing. ISBN: 1891121316.
- Pereyra, M. and H. Batatia (2012). "Modeling ultrasound echoes in skin tissues using symmetric Alpha-Stable processes". In: *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on* 59.1, pp. 60–72. ISSN: 0885-3010. DOI: [10.1109/TUFFC.2012.2156](https://doi.org/10.1109/TUFFC.2012.2156).
- Prager, R.W et al. (2003). "Decompression and speckle detection for ultrasound images using the homodyned k-distribution". In: *Pattern Recognition Letters* 24.4-5, pp. 705–713. DOI: [10.1016/S0167-8655\(02\)00176-9](https://doi.org/10.1016/S0167-8655(02)00176-9). URL: [http://dx.doi.org/10.1016/S0167-8655\(02\)00176-9](http://dx.doi.org/10.1016/S0167-8655(02)00176-9).

- Raju, B.I. and M.A. Srinivasan (2002). "Statistics of envelope of high-frequency ultrasonic backscatter from human skin in vivo". In: *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control* 49.7, pp. 871–882. DOI: [10.1109/tuffc.2002.1020157](https://doi.org/10.1109/tuffc.2002.1020157). URL: <http://dx.doi.org/10.1109/TUFFC.2002.1020157>.
- Rayleigh, Lord (1880). "On the resultant of a large number of vibrations of the same pitch and of arbitrary phase". In: *Philosophical Magazine Series* 5 10.60, pp. 73–78. DOI: [10.1080/14786448008626893](https://doi.org/10.1080/14786448008626893). URL: <http://dx.doi.org/10.1080/14786448008626893>.
- Rice, S. O. (1945). "Mathematical Analysis of Random Noise". In: *Bell System Technical Journal* 24.1, pp. 46–156. DOI: [10.1002/j.1538-7305.1945.tb00453.x](https://doi.org/10.1002/j.1538-7305.1945.tb00453.x). URL: <http://dx.doi.org/10.1002/j.1538-7305.1945.tb00453.x>.
- Shanbhag, D.N. and M. Sreehari (1977). "On certain self-decomposable distributions". English. In: *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 38.3, pp. 217–222. ISSN: 0044-3719. DOI: [10.1007/BF00537265](https://doi.org/10.1007/BF00537265). URL: <http://dx.doi.org/10.1007/BF00537265>.
- Shankar, P. Mohana (2000). "A general statistical model for ultrasonic backscattering from tissues". In: *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control* 47.3, pp. 727–736. DOI: [10.1109/58.842062](https://doi.org/10.1109/58.842062). URL: <http://dx.doi.org/10.1109/58.842062>.
- Shankar, P.M. (2001). "Ultrasonic tissue characterization using a generalized Nakagami model". In: *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control* 48.6, pp. 1716–1720. DOI: [10.1109/58.971725](https://doi.org/10.1109/58.971725). URL: <http://dx.doi.org/10.1109/58.971725>.
- (2003). "A compound scattering pdf for the ultrasonic echo envelope and its relationship to K and Nakagami distributions". In: *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control* 50.3, pp. 339–343. DOI: [10.1109/tuffc.2003.1193628](https://doi.org/10.1109/tuffc.2003.1193628). URL: <http://dx.doi.org/10.1109/TUFFC.2003.1193628>.
- Shankar, P.M. et al. (1993). "Use of non-Rayleigh statistics for the identification of tumors in ultrasonic B-scans of the breast". In: *IEEE Transactions on Medical Imaging* 12.4, pp. 687–692. DOI: [10.1109/42.251119](https://doi.org/10.1109/42.251119). URL: <http://dx.doi.org/10.1109/42.251119>.
- Shankar, P.M. et al. (2001). "Classification of ultrasonic B-mode images of breast masses using Nakagami distribution". In: *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control* 48.2, pp. 569–580. DOI: [10.1109/58.911740](https://doi.org/10.1109/58.911740). URL: <http://dx.doi.org/10.1109/58.911740>.
- Springer, M.D. (1979). *The algebra of random variables*. Probability and Statistics Series. Wiley. ISBN: 9780471014065. URL: <https://books.google.es/books?id=qUDvAAAAMAAJ>.
- Tsui, Po-Hsiang et al. (2008). "Classification of breast masses by ultrasonic Nakagami imaging: a feasibility study". In: *Physics in Medicine and Biology* 53.21, pp. 6027–6044. DOI: [10.1088/0031-9155/53/21/009](https://doi.org/10.1088/0031-9155/53/21/009). URL: <http://dx.doi.org/10.1088/0031-9155/53/21/009>.
- Tur, M., K. C. Chin, and J. W. Goodman (1982). "When is speckle noise multiplicative?" In: *Appl. Opt.* 21.7, pp. 1157–1159. DOI: [10.1364/AO.21.001157](https://doi.org/10.1364/AO.21.001157). URL: <http://ao.osa.org/abstract.cfm?URI=ao-21-7-1157>.
- Tuthill, T. A., R. H. Sperry, and K. J. Parker (1988). "Deviations from Rayleigh Statistics in Ultrasonic Speckle". In: *Ultrasonic Imaging* 10.2, pp. 81–89. DOI: [10.1177/016173468801000201](https://doi.org/10.1177/016173468801000201). URL: <http://dx.doi.org/10.1177/016173468801000201>.

- Twersky, Victor (1978). "Acoustic bulk parameters in distributions of pair-correlated scatterers". In: *The Journal of the Acoustical Society of America* 64.6, p. 1710. DOI: [10.1121/1.382150](https://doi.org/10.1121/1.382150). URL: <http://dx.doi.org/10.1121/1.382150>.
- (1987). "Low-frequency scattering by correlated distributions of randomly oriented particles". In: *The Journal of the Acoustical Society of America* 81.5, p. 1609. DOI: [10.1121/1.394513](https://doi.org/10.1121/1.394513). URL: <http://dx.doi.org/10.1121/1.394513>.
- (1988). "Low-frequency scattering by mixtures of correlated nonspherical particles". In: *The Journal of the Acoustical Society of America* 84.1, p. 409. DOI: [10.1121/1.396945](https://doi.org/10.1121/1.396945). URL: <http://dx.doi.org/10.1121/1.396945>.
- Twerskyt, Victor (1975). "Transparency of pair-correlated, random distributions of small scatterers, with applications to the cornea". In: *Journal of the Optical Society of America* 65.5, p. 524. DOI: [10.1364/josa.65.000524](https://doi.org/10.1364/josa.65.000524). URL: <http://dx.doi.org/10.1364/JOSA.65.000524>.
- Vegas-Sanchez-Ferrero, Gonzalo et al. (2012). "A Generalized Gamma Mixture Model for Ultrasonic Tissue Characterization". In: *Computational and Mathematical Methods in Medicine* 2012, pp. 1–25. DOI: [10.1155/2012/481923](https://doi.org/10.1155/2012/481923). URL: <http://dx.doi.org/10.1155/2012/481923>.
- Wagner, R.F. et al. (1983). "Statistics of Speckle in Ultrasound B-Scans". In: *IEEE Transactions on Sonics and Ultrasonics* 30.3, pp. 156–163. DOI: [10.1109/t-su.1983.31404](https://doi.org/10.1109/t-su.1983.31404). URL: <http://dx.doi.org/10.1109/T-SU.1983.31404>.
- Wagner, Robert F. (1986). "Unified Approach To The Detection And Classification Of Speckle Texture In Diagnostic Ultrasound". In: *Optical Engineering* 25.6. DOI: [10.1117/12.7973899](https://doi.org/10.1117/12.7973899). URL: <http://dx.doi.org/10.1117/12.7973899>.
- Wagner, Robert F., Michael F. Insana, and David G. Brown (1987). "Statistical properties of radio-frequency and envelope-detected signals with applications to medical ultrasound". In: *Journal of the Optical Society of America A* 4.5, p. 910. DOI: [10.1364/josaa.4.000910](https://doi.org/10.1364/josaa.4.000910). URL: <http://dx.doi.org/10.1364/JOSAA.4.000910>.
- Ward, K.D. (1981). "Compound representation of high resolution sea clutter". In: *Electronics Letters* 17.16, pp. 561–563. ISSN: 0013-5194. DOI: [10.1049/el:19810394](https://doi.org/10.1049/el:19810394).
- Xie, Hua, L.E. Pierce, and F.T. Ulaby (2002). "Statistical properties of logarithmically transformed speckle". In: *Geoscience and Remote Sensing, IEEE Transactions on* 40.3, pp. 721–727. ISSN: 0196-2892. DOI: [10.1109/TGRS.2002.1000333](https://doi.org/10.1109/TGRS.2002.1000333).